NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ATTITUDE DETERMINATION FOR THE THREE-AXIS SPACECRAFT SIMULATOR (TASS) BY APPLICATION OF PARTICLE FILTERING TECHNIQUES

by

Ioannis Kassalias

June 2005

Thesis Advisor:	Roberto Cristi
Second Reader:	David C. Jenn

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: Attitude Determination for the Three-Axis Spacecraft Simulator (TASS) by Application of Particle Filtering Techniques	**5. FUNDING NUMBERS**
6. AUTHOR(S) Ioannis Kassalias	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The accurate determination of spacecraft attitude has always been a critical issue in many applications. The presence of imperfect sensors introduces errors in the system and affects the outcome of the mission. One of the most significant sensors is the rate gyroscope. Particularly, the rate gyros are known to degrade with time, introducing random noise and bias. This calls for estimation algorithms which process the measured data in order to reduce the effects of the disturbances to a minimum. This research presents an approach which takes full advantage on the nonlinear dynamics and possibly non-Gaussian disturbances. It is based on recent work involving particle filters, where the probability density functions are approximated by a relatively large number of parameters. It is shown that accurate attitude estimation can be obtained with a manageable number of particles.

.

.

14. SUBJECT TERMS Particle Filter, Quaternion Representation, Attitude Determination, Rate Gyroscope, Nonlinear estimation	**15. NUMBER OF PAGES** 91
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

ATTITUDE DETERMINATION FOR THE THREE-AXIS SPACECRAFT SIMULATOR (TASS) BY APPLICATION OF PARTICLE FILTERING TECHNIQUES

Ioannis Kassalias
Lieutenant, Hellenic Navy
B.S., Hellenic Naval Academy, 1995

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
and
MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2005

Author: Ioannis Kassalias

Approved by: Roberto Cristi
Thesis Advisor

David C. Jenn
Second Reader

John P. Powers
Chairman, Department of Electrical and Computer Engineering

Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The accurate determination of spacecraft attitude has always been a critical issue in many applications. The presence of imperfect sensors introduces errors in the system and affects the outcome of the mission. One of the most significant sensors is the rate gyroscope. Particularly, the rate gyros are known to degrade with time, introducing random noise and bias. This calls for estimation algorithms which process the measured data in order to reduce the effects of the disturbances to a minimum. This research presents an approach which takes full advantage on the nonlinear dynamics and possibly non-Gaussian disturbances. It is based on recent work involving particle filters, where the probability density functions are approximated by a relatively large number of parameters. It is shown that accurate attitude estimation can be obtained with a manageable number of particles.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First of all, I am grateful to my beloved wife, Mara, who treated me with all of her patience and understanding throughout my efforts on accomplishing my degree. The fact that during our stay in Monterey she also delivered my divine daughter, Stella-Zoe, the best present I ever received in my whole life, took care of her in an amazingly lovesome way and achieved her own degree makes her effort inexpressible.

I would also like to dedicate this thesis to my supportive parents who did their best to grow me up in an ethical and productive way.

I would like to thank my thesis advisor, Professor Roberto Cristi, whose excellent guidance and instruction helped me on my way to thesis completion and Professor David C. Jenn who also contributed to it as my second reader. A special thank you is also in order of Dr. Jong-Woo Kim who patiently helped me find my way through the required knowledge of the spacecraft simulator.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The great importance of military and civilian space systems has driven the need for extending their expected lifetime and increasing the associated components' accuracy. The rate gyroscope is one of the vital sensors on a spacecraft and its performance dramatically affects both lifetime and performance. The reduction of the rate gyro noise level has been an issue for quite a long time.

In order to mitigate the undesirable effects of bias and measurement noise in a rate gyroscope, various techniques have been developed. This thesis investigates how a new approach can be utilized in order to achieve better accuracy in a strongly nonlinear system such as a spacecraft attitude control system.

At first, the particle filtering technique is discussed and compared to traditional techniques such as Kalman filtering and extended Kalman filtering. Based on the kind of model and the type of the disturbances, the expected performance of each approach is analyzed. Initially, in order to understand the proposed algorithm, particle filter techniques and the Kalman filter are compared using a simple dynamic model. Then the results for one-axis rotation of the spacecraft simulator are presented in terms of attitude estimation.

The first model of interest is a linear model in Gaussian environment. The particle filter is compared to the Kalman one which, as expected, is shown to be the best estimator for this specific case. Through a series of simulations, the number of required particles is inferred.

The second model is the same linear one, but in additive non-Gaussian noise. In this case the Kalman filter can not guarantee convergence, so other techniques must be developed. The scope of the associated simulations is to show the particle filter robustness when enough particles are propagated. On the other hand, when the number of particles is insufficient, large errors occur in the estimation procedure.

Finally, the particle filter is evaluated for the specific application of spacecraft attitude determination using a quaternion-based representation of one-axis rotation when Gaussian noise ia added in both the state and measurement equations of the developed nonlinear model. The parameters were chosen in a way that the model included discontinuities that had to be overcome by the proposed estimator. The extended Kalman filter cannot provide a direct solution to the problem since it is based on local linearization of the nonlinear model, which can only be performed in continuous processes.

The particle filter proved to be an excellent estimator for this particular application. The objective was to minimize the mean-square error of the measured roll angle. The results showed that a particle filter using 2000 particles can provide a significant improvement with respect to the roll angle measurement error. This conclusion, in conjunction with the increasing computational power of today's machines, indicated that the particle filter can be widely utilized in this and similar applications.

I. INTRODUCTION

There has been a considerable evolution in the development of space systems during the past few decades. One of the key aspects is platform stabilization and navigation. The challenge is to keep the space platform at a specific orientation with respect to a fixed inertial reference frame. In particular, requirements can become very strict in some applications. For example, in laser pointing applications errors on the order of one millidegree could result in a target miss by several kilometers.

The general approach to reducing the impact of such errors is to combine the observations from a number of sensors which include star trackers and gyroscopes measuring angular rates. The dynamic equations relating the measured quantities to the platform orientations are nonlinear. The presence of miscellaneous errors due to various sensors increases the complexity of the problem. This thesis investigates a new approach and evaluates its effectiveness.

Kalman filtering applications to spacecraft attitude estimation were introduced in the early 1970s. Since then, many advanced algorithms based on the Kalman filter have been developed and implemented with varying levels of success. Most of them are based on local linearization of nonlinear processes in order to apply filtering. Development of a purely nonlinear technique that is applicable to spacecraft attitude determination and overcomes the problems of singularities and discontinuities is an extremely important issue.

A. OBJECTIVE

The objective of this thesis was the discretization, implementation and evaluation of an algorithm for attitude determination that compensates for external disturbances in the gyroscopes. This algorithm was based on a newly developed technique that fully exploits nonlinearities. In particular, an approach based on particle filtering was evaluated in various cases. Particle filter applications have already been developed in fields such as wireless communications, target tracking and navigation systems. Some of them can be found in the list of references.

B. THESIS OUTLINE

The thesis starts with an overview of the various aspects of the estimation problem, and existing nonlinear algorithms are analyzed briefly. Different variations of the particle filter are presented along with background information on rigid body kinematics. Next, dynamic modeling of a rate gyro and the related algorithm development are discussed. Finally, the simulation results are presented and summarized.

In particular, the thesis is organized as follows:

Chapter II provides a survey on the existing filtering algorithms, the particle filter and its application to rigid body kinematics. The parameters used in this thesis are defined and explained in this chapter, as well.

Chapter III describes the target system, which is the NPS Three-Axis Spacecraft Simulator (TASS). The development of the proposed dynamic model and its discrete time implementation for this system is also presented here.

Chapter IV presents the performance analysis and evaluation of the particle filter algorithm for all cases that it was tested.

Chapter V summarizes the conclusions and provides recommendations for the future research.

II. BACKGROUND INFORMATION

In this chapter, the theoretical background for this thesis is presented to the reader. In particular, the general estimation problem is established and the associated estimation techniques are analyzed, emphasizing on the particle filter. Finally, an introductory discussion on the rigid body attitude representation concludes the chapter.

A. THE GENERAL ESTIMATION PROBLEM

In a number of applications we need to estimate a particular state of the system, based on observations from sensors. A mathematical model relates the state dynamics to the measured signals and the measurement errors. This model, also called the *dynamic* model, can be completely described by the following set of equations:

$$\dot{x}(t) = f\left[x(t)\right] + w(t) \tag{1}$$

$$y(t) = g\left[x(t)\right] + v(t) \tag{2}$$

where $x(t)$ is the state to be estimated, $y(t)$ is the available set of measurements, f and g are functions related to the state, $w(t)$ is the state noise and $v(t)$ is the measurement noise. In this general case, the functions f and g can be of any form and the statistics of the noise signals are assumed to be known. The state noise signal $w(t)$ is a measure of the accuracy for the dynamic model, while the measurement noise signal $v(t)$ is directly related to the specifications of the sensors. A basic assumption for the model is that these two noise signals are uncorrelated. Equation (1) is called the state equation and presents the evolution of the state in time, while Eq. (2) is called the measurement or observation equation and is the one that relates the state to the measurement.

The model described above is a continuous time one. However, since most of the applications are based on sampled data, discrete time models are widely used. This type of model can be completely described by the following set of equations:

$$x_{k+1} = f\left[x_k\right] + w_k \tag{3}$$

$$y_k = g\left[x_k\right] + v_k \tag{4}$$

where *x* is the state to be estimated, *y* is the available set of measurements, *f* and *g* are functions related to the state, *w* and *v* are uncorrelated noise sequences, and finally *k* is the index or iteration number. In this discrete time model, the state equation represents the evolution of the state vector while the measurement equation relates the state to the measurement at each sample point.

In each application the estimation problem consists of two basic stages, mathematical modeling and state estimation. Mathematical modeling includes the derivation of the functions *f* and *g* based on physical considerations. The statistics of the noise, assumed Gaussian for convenience, are in general difficult to determine and greatly influence the performance of the estimator. The estimation stage includes an algorithm that provides the best estimate of the state given a set of measurements and any initial conditions and constraints that apply.

The general estimation problem can be divided in two categories, linear and nonlinear. Also, with respect to the statistics of the comprised noise, discrimination between Gaussian and non-Gaussian noise is useful.

Due to the difficulty of the general problem, most applications have been based on simplifying assumptions. The easiest approach is based on a linear model and Gaussian disturbances. This leads to the utilization of the Kalman filter introduced in the next section.

B. THE LINEAR CASE IN GAUSSIAN ENVIRONMENT

1. System Modeling

A significant number of tracking applications can be implemented using linear models in a Gaussian environment. This specific type of model can be described by the following set of equations in discrete time:

$$x_{k+1} = F_k x_k + w_k \tag{5}$$

$$y_k = H_k x_k + v_k \tag{6}$$

where F_k is the matrix relating x_k to x_{k+1} in the absence of a forcing function (similar to the state transition matrix for continuous time processes), H_k is the matrix corresponding to the relationship between the measurement and the state vector, w_k and v_k are white Gaussian uncorrelated noise sequences with covariance Q_k and R_k, respectively, while index *k* refers to the time step t_k. Assuming that *x* is an *n*-dimensional vector while *y* is an *m*-dimensional one, F_k has to be of dimension $n \times n$ and H_k of $m \times n$. Both F_k and H_k can be time variant provided their values are known for each time step. We can observe that this model is determined by only linear relations. Also, noise sequences can be fully described by two values, mean and covariance, and they can be time variant as well, assuming that the covariance values are known for each time step. The following equations fully describe the statistical properties of the noise sequences:

$$\mathrm{E}\left\{w_k w_i^T\right\} = Q_k \delta\left[k-i\right] \tag{7}$$

$$\mathrm{E}\left\{v_k v_i^T\right\} = R_k \delta\left[k-i\right] \tag{8}$$

$$\mathrm{E}\left\{w_k v_i^T\right\} = 0 \quad \forall\, i,k \in \mathbb{R} \tag{9}$$

where $\mathrm{E}\{.\}$ is the expected value operator, Q_k and R_k are the covariance values of the noise sequences w_k and v_k respectively, and δ is the Dirac function.

2. The Kalman Filter

A recursive algorithm providing solution to this kind of problems was developed by R.E. Kalman in 1960 and it is known as the Kalman filter [1]. As it is stated in the previous chapter, the objective is to estimate the state for each time step. For this reason, the state estimate at time t_k is denoted by $\hat{x}_k$ and a new variable called the estimation error is defined by

$$e_k = x_k - \hat{x}_k. \tag{10}$$

The error covariance matrix is now defined by

$$P_k = E\left\{e_k e_k^T \mid \mathbf{y}_k\right\} \tag{11}$$

where $\mathbf{y}_k = \left\{y_i, i=1,...,k\right\}$ is the vector containing all the available measurements up to time step t_k.

It is obvious that the estimation error vector is *n*-dimensional, so the error covariance matrix is of dimension $n \times n$. The problem now leads to the minimization of e_k, which is equivalent to the minimization of the trace of P_k. The Kalman filter accomplishes this task by following the procedure indicated in Fig. 1. An overview of the intermediate stages is presented in the following sections.

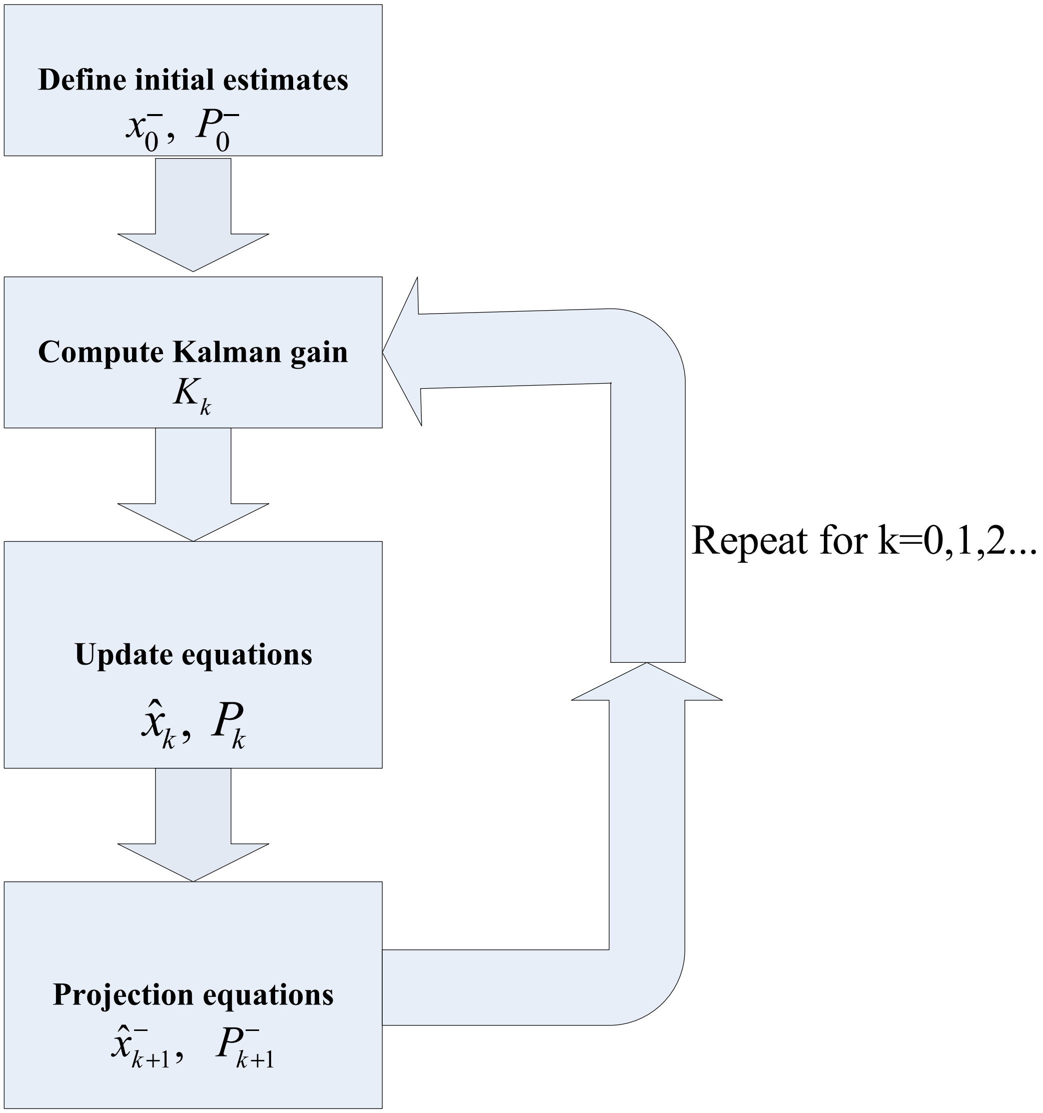


Figure 1. Block Diagram of the Kalman Filter

a. Initialization

The Kalman filter is a recursive algorithm and it is repeated at each time step. However, initial estimates for the state vector and the error covariance have to be determined. These values are denoted by $\hat{x}_0^-$ and P_0^- where the minus sign in the superscript indicates that the related measurement has not yet been taken under consideration. If no estimate for the initial value of the state is available, then it can be assumed that

$$\begin{aligned} \hat{x}_0^- &= 0 \\ P_0^- &= Q_0. \end{aligned} \tag{12}$$

The Kalman filter is guaranteed to converge to the actual value of the state, no matter what value of initial estimate is used. The error of the initial estimate affects only the number of iterations needed to reach convergence.

b. Calculation of the Kalman Gain

Knowing the posterior estimate $\hat{x}_k^- = E\{x_k \mid \mathbf{y}_{k-1}\}$ for each time step t_k, the measurement at that time has to be taken into account. This happens by taking advantage of the following equation:

$$\hat{x}_k = (I - K_k H_k)\hat{x}_k^- + K_k y_k \tag{13}$$

where *I* is the identity matrix and the quantity K_k is the Kalman gain. From Eq. (13) it is obvious that the Kalman gain is a weighting factor for both the measurement and *a priori* estimate, and it is related to the reliability of the observation. The optimal Kalman gain is constrained by the minimization criterion for the trace of the error covariance matrix. This leads to the following equation that provides the optimal Kalman gain:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}. \tag{14}$$

c. Update Equations

In this stage the measurement is taken into account and an update to both state estimation and error covariance takes place. The optimal error covariance matrix is obtained using the Kalman gain computed in the previous stage. The equations that drive this stage are the following:

$$\hat{x}_k = \hat{x}_k^- + K_k\left(y_k - H_k \hat{x}_k^-\right) \tag{15}$$

$$P_k = (I - K_k H_k) P_k^-. \tag{16}$$

d. Projection Equations

The update stage concludes the estimation for time step t_k. However, predictions for both state vector and error covariance matrix have to be generated in order to continue the recursion to the following time step t_{k+1}. These predictions depend only on the current state and carry no information based on the next time step. The equations providing these predictions are called projection equations and are as follows

$$\hat{x}_{k+1}^{-} = F_k \hat{x}_k \tag{17}$$

$$P_{k+1}^{-} = F_k P_k F_k^T + Q_k . \tag{18}$$

Once this stage is completed, recursion continues with the calculation of the Kalman gain for the next time step.

C. THE NONLINEAR CASE

The Kalman filter has been theoretically proven to be the best estimator for the linear case estimation problem described in the previous section [1], [2]. However, a number of current applications include nonlinearities and additive non-Gaussian noise which makes the Kalman filter inadequate. This has led to the development of advanced algorithms fitted to the requirements of each specific case.

1. The General Nonlinear Case

The general form of the nonlinear case model can be specified by Eqs. (3) and (4) under the assumption that either or both functions *f* and *g* are nonlinear. It is obvious that Kalman filter cannot be directly applied in this kind of model. Several techniques have been developed to overcome the difficulties introduced by the nonlinearities, considering the specific task to be accomplished. An overview of the most commonly used ones is the motivation for the following sections.

2. The Extended Kalman Filter (EKF)

The EKF is used in various real-time applications due to the fact that it is less computationally intensive when compared to other nonlinear estimation techniques. It is a regular Kalman filter that uses local linearization based on a first-order Taylor series expansion about the current estimate of the state and its covariance matrix. Let the model of interest be the one specified by Eqs. (1) and (2) and both noise sequences w_k and v_k have the same properties as described in the linear case. The difference $\Delta x(t)$ between the actual and the estimated value can be introduced and used in the model as described by the following set of equations:

$$\dot{\hat{x}}(t) + \Delta\dot{x}(t) = f\left[\hat{x}(t) + \Delta x(t)\right] + w(t) \tag{19}$$

$$y(t) = g\left[\hat{x}(t) + \Delta x(t)\right] + v(t). \tag{20}$$

Assuming that $\Delta x(t)$ is small, an approximation using a Taylor-series expansion along with the fact that $\dot{\hat{x}}(t) = f\left[\hat{x}(t)\right]$ yields:

$$\dot{\hat{x}}(t) + \Delta\dot{x}(t) = f\left[\hat{x}(t)\right] + \frac{\partial f}{\partial x}\Delta x(t) + w(t) \Rightarrow \Delta\dot{x}(t) = \frac{\partial f}{\partial x}\Delta x(t) + w(t) \tag{21}$$

$$y(t) - g\left[\hat{x}(t)\right] = \frac{\partial g}{\partial x}\Delta x(t) + v(t). \tag{22}$$

The quantity $\Delta x(t)$ can be treated as the state, and the quantity $y(t) - g\left[\hat{x}(t)\right]$ can be treated as the measurement, to provide a linear model where the Kalman filter can be directly applied. The discrete time equivalent system uses the model described by Eqs. (3) and (4) where

$$F_k = \left.\frac{\partial f\left[x_k\right]}{\partial x_k}\right|_{\hat{x}_k} \tag{23}$$

$$H_k = \left.\frac{\partial g\left[x_k\right]}{\partial x_k}\right|_{\hat{x}_k}. \tag{24}$$

The procedure to be followed is the same as the one described in the linear case. A block diagram indicating the calculations that are to be executed step by step is shown in Fig. 2.

The EKF described in this section uses a first-order linearization since only the first term of the Taylor-series expansion is used. More accurate results may be obtained using a higher-order EKF that retains higher-order terms of the Taylor series, but the computational intensity is significantly increased.

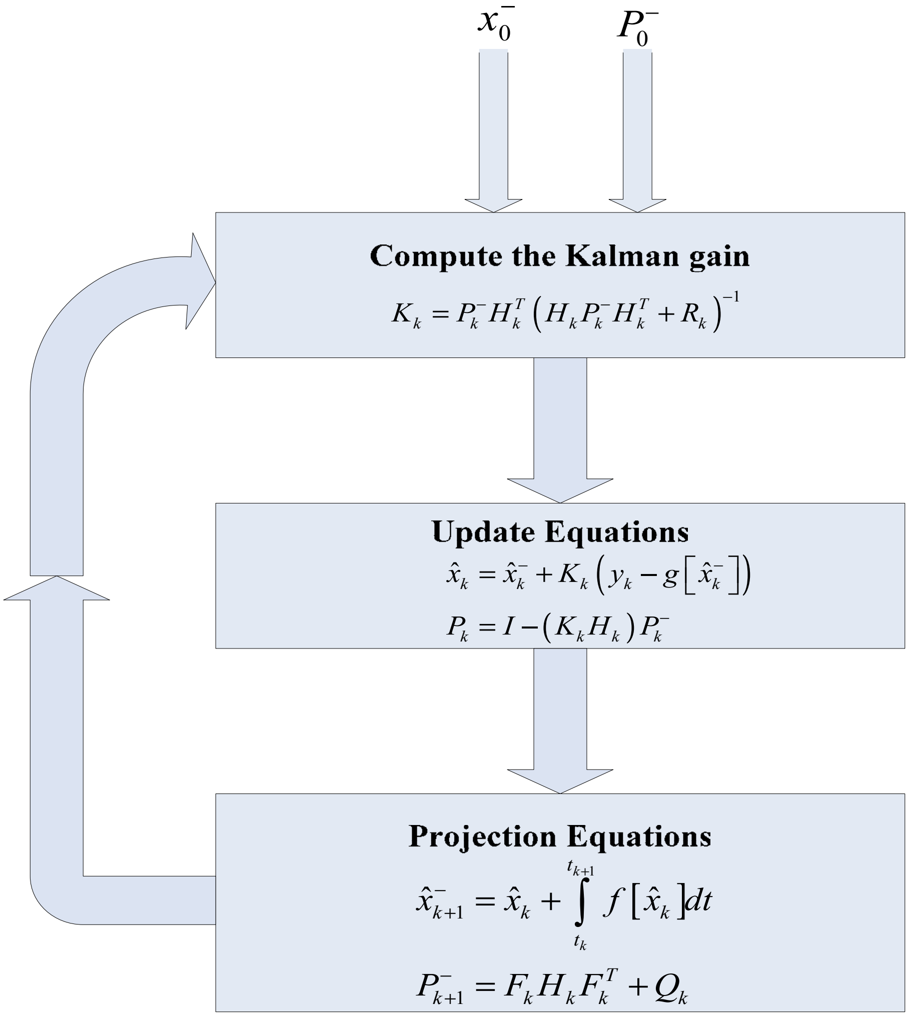

x_0^-
P_0^-
Compute the Kalman gain
$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1}$
Update Equations
$\hat{x}_k = \hat{x}_k^- + K_k \left(y_k - g\left[\hat{x}_k^- \right] \right)$
$P_k = I - \left(K_k H_k \right) P_k^-$
Projection Equations
$\hat{x}_{k+1}^- = \hat{x}_k + \int_{t_k}^{t_{k+1}} f\left[\hat{x}_k \right] dt$
$P_{k+1}^- = F_k H_k F_k^T + Q_k$

Figure 2. Block Diagram of the EKF

2. The Grid-Based Methods

In order to proceed to the analysis of the grid-based methods, first a statistical approach of the estimation problem is presented. The goal remains to provide a filtered estimate of the state x_k at time step *k* given the available set of measurements $\mathbf{y}_k = \{y_i, i = 1, ..., k\}$ up to time step *k*. From a Bayesian perspective, the problem is to recursively determine a measure of belief in the state x_k at time step *k*, taking different values and utilizing the knowledge of the available set of measurements $\mathbf{y}_k$ [2]. Hence, the posterior probability density function (pdf) $p\left(x_k \mid \mathbf{y}_k\right)$ is constructed and updated in each time step. The whole procedure can be described as follows [2], [3].

The initial density of the state vector, $p(x_0)$, is assumed to be known and independent of any measurement. Estimation of the pdf in each time step is obtained recursively, implementing sequentially the prediction and update stage. Assuming that the pdf $p\left(x_{k-1} \mid \mathbf{y}_{k-1}\right)$ in time step $k-1$ is available, the estimation of the pdf at the next time step without knowledge of the related measurement (that is the prediction) is given by

$$p\left(x_k \mid \mathbf{y}_{k\text{-}1}\right)=\int p\left(x_k \mid x_{k-1}\right) p\left(x_{k-1} \mid \mathbf{y}_{k\text{-}1}\right) d x_{k-1} \tag{25}$$

where the pdf $p\left(x_k \mid x_{k-1}\right)$ is fully defined based on the state Equation (3) and the statistics of the process noise sequence w_k. The update stage evolves the prediction by utilization of the next measurement y_k and updating by using:

$$p\left(x_k \mid \mathbf{y}_k\right)=p\left(x_k \mid y_k, \mathbf{y}_{k\text{-}1}\right)=\frac{p\left(y_k \mid x_k\right) p\left(x_k \mid \mathbf{y}_{k\text{-}1}\right)}{p\left(y_k \mid \mathbf{y}_{k\text{-}1}\right)} \tag{26}$$

where the normalizing pdf $p\left(y_k \mid \mathbf{y}_{k\text{-}1}\right)$ is fully defined on the basis of the measurement Equation (4) and the statistics of the measurement noise v_k.

For the grid-based methods the discretization of the above formulation is used. Assuming that the state space consists of N discrete states at time step $k-1$ denoted by x_{k-1}^i, $i=1,\ldots,N$ and the conditional probability of that state at the same time step is termed as $\omega_{k-1|k-1}^i$, the posterior pdf can be written as:

$$p\left(x_k \mid \mathbf{y}_k\right)=p\left(x_{k-1} \mid \mathbf{y}_{k\text{-}1}\right)=\sum_{i=1}^N \omega_{k-1|k-1}^i \delta\left(x_{k-1}-x_{k-1}^i\right) \tag{27}$$

Substitution of Eq. (27) into Eqs. (25) and (26) provides the prediction and update equations for the grid-based methods:

$$p\left(x_k \mid \mathbf{y}_{k\text{-}1}\right)=\sum_{i=1}^N \omega_{k|k-1}^i \delta\left(x_k-x_k^i\right) \tag{28}$$

where $\omega_{k|k-1}^i=\sum_{j=1}^N \omega_{k-1|k-1}^j p\left(x_k^i \mid x_{k-1}^j\right)$ and

$$p\left(x_k \mid \mathbf{y}_k\right) = \sum_{i=1}^{N} \omega_{k|k}^{i} \delta\left(x_k - x_k^i\right) \tag{29}$$

where $\omega_{k|k}^{i} = \omega_{k|k-1}^{i} p\left(y_k \mid x_k^i\right) \Big/ \sum_{j=1}^{N} \omega_{k|k-1}^{j} p\left(y_k \mid x_k^j\right)$.

The recursive propagation of the pdf that is described by Eqs. (28) and (29) provides enough information to proceed to the optimal state estimate, depending on the criterion that is established for any case. That is, if the minimum mean square error (MMSE) is the criterion, then the estimate is given by [2]:

$$\hat{x}_{k|k}^{MMSE} = E\left\{x_k \mid \mathbf{y}_k\right\} = \int x_k p\left(x_k \mid \mathbf{y}_k\right) dx_k. \tag{30}$$

In the case of the maximum a posteriori (MAP) estimate, the maximum value of the pdf is used [2]:

$$\hat{x}_{k|k}^{MAP} = \arg\max_{x_k} \left(x_k \mid \mathbf{y}_k\right). \tag{31}$$

The use of the grid-based methods is constrained by the assumption that the state space has to be discretized in a finite number of states [2]. These constraints, along with the extremely high computational cost in multi-dimensional state space cases and the requirement for sufficiently dense grid [3], significantly restricts the number of applications to which this technique can be applied.

3. The Particle Filter

The concept of particle filtering was originally proposed back in 1954 [2] in the form of the Sequential Monte Carlo (SMC) estimation based on particle representation of probability densities. However, despite the continuous theoretical exploration of the initial idea during the 1960s and 1970s, no effort to apply it to a specific application took place. The main constraint obviously was the amount of computational effort that is required to apply such a technique. The fact that the computational power of the machines at that time was quite poor leads to the conclusion that the implementation of a complete system using the proposed technique was not feasible. Since computational power has been tremendously increased during the past decades, the above mentioned constraint has been eliminated. Particle filter applications in various fields such as wireless communications [4], target tracking [5], and navigation systems [6] have recently been proposed.

a. Overview of the Particle Filter (PF)

The idea of the PF technique is based on the statistical approach of the nonlinear estimation problem discussed previously. The main feature is the representation of the density function of interest using a random set of samples for the state value with associated weights. This kind of representation recursively leads to the estimation of the pdf for the next time step, taking into account the measurement associated with it. If the number of samples used in each iteration is sufficient, the representation of the pdf is quite accurate and it provides good estimates.

Let $\mathbf{x}_k$ represent the set of the target values of the state for time step 0 to *k*. Assuming that in each time step a number of *N* samples are drawn, the set $\left\{\mathbf{x}_k^i, \omega_k^i\right\}_{i=1}^N$ of the support points with the associated weights fully characterizes the posterior pdf $p\left(\mathbf{x}_k \mid \mathbf{y}_k\right)$. The weights are selected in such a way that the weighting vector is normalized, that is $\sum_{i=1}^{N} \omega_k^i = 1$ for each time step *k*. Under these assumptions, Eq. (29) may be used in order to provide an approximation of the posterior filtered density $p\left(x_k \mid \mathbf{y}_k\right)$.

b. The Sequential Importance Sampling (SIS) Algorithm

The basis of each PF technique is the SIS algorithm (or otherwise called *Bootstrap Filtering* [7]). According to this algorithm the particles, meaning the state samples, are propagated in each iteration by utilization of two stages, the prediction and the update.

During the prediction stage, *N* samples of the process noise w_k are drawn according to the statistics of the noise sequence. Then, each sample point of the state is propagated using the state equation, without taking into account the corresponding importance weights. This provides a posterior prediction of the particles for the next time step according to the equation

$$x_{k+1}^i = f_k(x_k^i, w_k^i) \tag{32}$$

where $i = 1,\ldots,N$ represents the index number of the sample and f_k is the function related to the state at time step *k* as defined in Eq. (3).

During the update stage the important weights are updated using the information provided by the measurement sequence up to time step *k*. The related equation is

$$\omega_{k+1}^{i} = \omega_{k}^{i} p(y_{k+1} \mid x_{k+1}^{i}) \tag{33}$$

where the likelihood function $p(y_{k+1} \mid x_{k+1}^{i})$ depends on the measurement equation introduced by Eq. (4) and the statistics of the measurement noise v_k. Before finalizing this stage, the weighting vector has to be normalized and a new value is applied to each weight using the equation

$$\omega_{k+1}^{i} = \frac{\omega_{k+1}^{i,o}}{\sum_{i=1}^{N} \omega_{k+1}^{i,o}} \tag{34}$$

where the superscript *o* indicates the weight value originally computed using Eq. (33) and ω_{k+1}^{i} is the weight value of the *i-th* particle to be propagated to the next iteration.

c. The Resampling Procedure – Sampling Importance Resampling (SIR) Algorithm

Unfortunately, in practice, the SIS algorithm has a significant drawback. After a certain number of steps, only one particle with significant associated weight value is left [7]. The rest of the weights are negligible leading to two unwanted effects, low accuracy and a large computational effort to propagate particles of negligible significance regarding their contribution to the approximation of the desired pdf $p\left(x_k \mid \mathbf{y}_k\right)$. This fact is known as the *degeneracy problem* [2] and its existence implies the need for an additional stage, resulting in a more robust algorithm than the SIS one.

The key point to the solution of the degeneracy problem is the definition of a new parameter called the effective number of samples, which is estimated by the relationship [3]:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} \left(\omega_{k}^{i}\right)^{2}} \tag{35}$$

where ω_k^i are the normalized values of the weights obtained at the end of the update stage using Eq. (34). Whenever $\hat{N}_{eff}$ has a small value, the degeneracy phenomenon is severe and a new stage called *resampling* has to be executed. During this stage, particles with negligible importance weights are discarded and replaced by samples with large weights. The importance weight of each remaining particle is a measure of its probability of repetition. Excluding the extreme cases of a uniform distribution of the importance weights ($\hat{N}_{eff} = N$) and of $N-1$ zero weights ($\hat{N}_{eff} = 1$), the resampling procedure provides an effective scheme in avoiding degeneracy.

The last parameter that has to be determined in order to apply resampling is the decision threshold for the effective number of samples. There is no general guideline for its selection. However, it has to be taken into account that a small threshold leads to high resampling ratio implying high computational cost, while a large value of the decision threshold leads to non-effective resampling because in each iteration some particles with negligible weights are reproduced.

The algorithm, including the resampling stage where necessary, is called *Sampling Importance Resampling (SIR).* In this thesis, a PF based on this algorithm is developed and evaluated.

D. ATTITUDE REPRESENTATION

In this section a short description of the available attitude representation methods along with the basic of rigid body kinematics equations are presented. The quaternion representation and the related kinematics equations are emphasized due to their extensive use in the modeling and simulation of the system described in Chapter IV.

The attitude representation of a rigid body can be described in terms of rotating the orientation of the body coordinate system to align with another one. The following methods can be applied to describe these kinds of rotations.

1. The Euler Angles

Euler angles define a series of three successive rotations around the body coordinate system in order to achieve the desired orientation [8]. Each rotation occurs around the original body axes. Since there are twelve possible ways to arrange sequentially the three axes, there are twelve different possible set of Euler angles. In this thesis the orientation roll φ (rotation about the x axis), pitch θ (rotation about the y axis) and yaw ψ (rotation about the z axis) is used. In this case a rotation can be represented by the following set of rotation matrices:

$$\begin{gathered}
C_1(\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
C_2(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \\
C_3(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix}.
\end{gathered} \tag{36}$$

The product of these three matrices is the rotation matrix between the initial and final orientations of the body and is a function of all three Euler angles.

2. Quaternion Representation

The quaternion $\mathbf{q}$ is a four-element vector which fully describes a rotation in a three dimensional space. The first element of the quaternion is a scalar value while the remaining three elements are vector components. The scalar value represents a measure of the magnitude of rotation. The remaining three elements are proportional to the *Euler axis* $\vec{e}$ around which the rotation occurs [8]. Letting θ_e be the angle of rotation, the quaternion components are defined as follows [9]:

$$\begin{aligned}
q_0 &= \cos\frac{\theta_e}{2} \\
q_1 &= e_x \sin\frac{\theta_e}{2} \\
q_2 &= e_y \sin\frac{\theta_e}{2} \\
q_3 &= e_z \sin\frac{\theta_e}{2}
\end{aligned} \tag{37}$$

where e_x, e_y and e_z are the three normalized to unitary magnitude elements of the Euler axis with respect to *x*, *y* and *z* axes, respectively.

A very important property of the quaternions is the normalization constraint which is determined by

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \tag{38}$$

The above equation means that the four elements of the quaternion vector are mutually dependent. This makes sense since a four-dimensional vector is used to describe a rotation in a three-dimensional space. The reason that the quaternion representation is extensively used in attitude representation is that it does not introduce singularities [10] and requires less computational effort to propagate when compared to representations including rotation matrices. However, the mapping between quaternions and Euler angles is not unique [11], [12] since any pair of quaternions $\mathbf{q}$ and $-\mathbf{q}$ represent exactly the same rotation. On the other hand, the *Gibbs vector* provides a three-dimensional one to one representation of a rotation vector.

3. The Gibbs Vector

The Gibbs vector **g** is a representation directly related to the quaternions by the following relationship:

$$\mathbf{g} = \frac{1}{q_0}\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{\alpha_g}{2}. \tag{39}$$

In order to understand the relation between the quaternion and the Gibbs vector, recall that the quaternion vector is defined on a four-dimensional sphere [11]. The Gibbs vector can be considered as a gnomonic projection of the quaternion vector from the sphere to the vector ***g*** as shown in Fig. 3. The use of the Gibbs vector eliminates the 2:1 mapping issue of the quaternion representation. However, it is clear that it generates a singularity for 180° rotations which limits its use in representations of rotations in the interval $(-\pi, \pi)$. Hence, the use of the Gibbs vector is not adequate when global representation of rotations is desired.

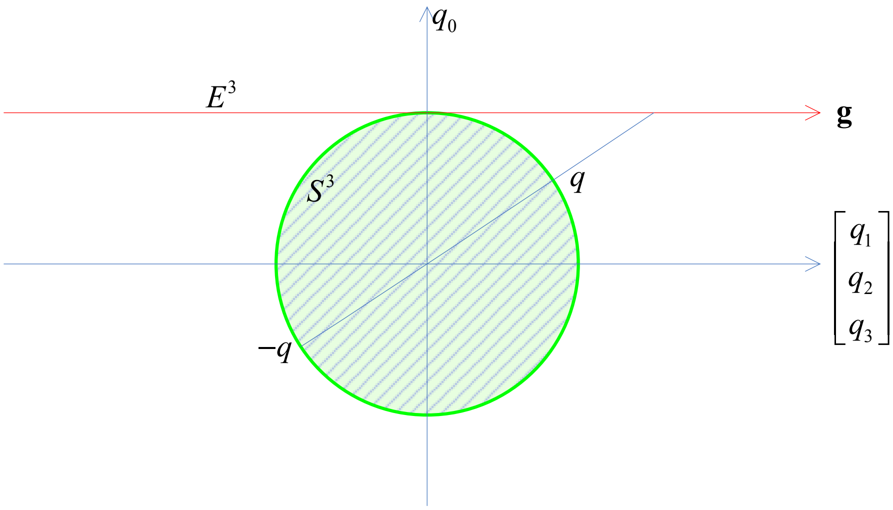


Figure 3. Gibbs vector as a gnomonic projection (After Ref. [11].)

4. Attitude Kinematics

In order to propagate the quaternion vector, an introduction to the quaternion kinematics equation is necessary. This equation describes the evolution of the quaternion vector in time with respect to the angular rate given by the following relationship:

$$\dot{\mathbf{q}} = \frac{1}{2}\Omega(\boldsymbol{\omega})\mathbf{q} \tag{40}$$

where $\boldsymbol{\omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$ is the angular rate vector of the body and the matrix Ω is of 4×4 dimension and it can be calculated by the equation:

$$\Omega(\boldsymbol{\omega}) = \begin{bmatrix} -[\boldsymbol{\omega}\times] & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix} \tag{41}$$

where $[\boldsymbol{\omega}\times]$ denotes the cross product matrix of the angular rate vector, which in the body coordinate system is given by:

$$[\boldsymbol{\omega}\times] = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{42}$$

By discretizing Eq. (40) with small time increments and utilizing Eqs. (41) and (42), a discrete-time dynamic model for the quaternion propagation can be derived.

In this chapter, various solutions to the estimation problem were discussed. The use of PF is of particular interest. In order to apply particle filtering, a dynamic model based on the attitude kinematics that were previously presented has to be developed. The derivation of such a model for the NPS TASS is the main point to be addressed in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DYNAMIC MODELING FOR THE NPS TASS

The objective of this chapter is the determination of a dynamic model for the NPS TASS. In particular this model is based on the quaternion representation and it is used both for simulation of the platform and the estimation by particle filtering. At first, an overview of the TASS is presented, then the rate gyroscope along with the associated disturbances is discussed. Finally, the dynamic model based on the attitude kinematics presented in the previous chapter, is developed.

A. SYSTEM OVERVIEW

The NPS TASS is a platform which can simulate the three-axis motion of a spacecraft. It is supported on a spherical air bearing which provides a frictionless motion in order to emulate a space environment. Basically, a thin film of air is trapped between the lower spherical end of the simulator and a semi-spherical cup of matched size which is mounted on the ground. A detailed description of the system is found in Refs. [13] and [14].

Various control components and sensors are installed in the TASS in order to provide enough data for simulation purposes. Three variable-speed control moment gyroscopes, rate gyros, two-axis analog sun sensors, an IR sensor and two inclinometers are included. In this thesis the rate gyros are of particular interest and they are described in terms of their dynamics and disturbances.

B. RATE GYROSCOPES

Rate gyroscopes are generally used to provide angular rate data to the spacecraft control system. Different types of rate gyroscopes, such as mechanical and laser, have been developed. Regardless of the type of the gyroscope, the task to be accomplished is to sense angular rate measurements of the satellite in all three dimensions and with respect to the satellite body coordinates.

There are two different types of errors included in a set of noisy measurements provided by a rate gyroscope, the bias and the noise. The main reasons for the presence of these errors are the manufacturing imperfections, misalignment and the degradation of the gyroscope through time. The measured value of the angular rate vector provided by the rate gyroscopes can be modeled using the equation:

$$\boldsymbol{\omega}_m(t) = \boldsymbol{\omega}_t(t) + \mathbf{b}(t) + \mathbf{n}(t) \tag{43}$$

where vector notation is used to indicate a three-dimensional quantity, $\boldsymbol{\omega}_m$ indicates the measured value, $\boldsymbol{\omega}_t$ indicates the true value, $\mathbf{b}(t)$ is the bias vector and $\mathbf{n}(t)$ is the three-dimensional noise vector consisting of independent white-noise sequences.

C. DERIVATION OF THE DYNAMIC MODEL

The availability of a set of measurements of the angular rate vector allows the generation of a part of the state equation using Eqs. (40) and (43). From them, an expression of the quaternion vector in terms of the measured values of the angular rate can be generated:

$$\dot{\mathbf{q}}(t) = \frac{1}{2}\Omega\left[\boldsymbol{\omega}_m(t) - \mathbf{b}(t) - \mathbf{n}(t)\right]\mathbf{q}(t) \tag{44}$$

where Ω is the matrix presented in Eq. (41).

The presence of a slowly varying bias can be modeled as a random walk stochastic process. This results in the differential equation:

$$\dot{\mathbf{b}}(t) = \mathbf{w}(t) \tag{45}$$

where $\mathbf{w}(t)$ is a three-dimensional noise vector consisting of independent white noise sequences and is uncorrelated to the noise vector $\mathbf{n}(t)$ referred above. This statistical property of the bias forces its inclusion as part of the state vector in the dynamic model.

As far as the measurement equation of the model is concerned, the combined measurements of the two inclinometers and the IR sensor can be utilized to provide a set of measured Euler angles. Since the quaternion is part of the state, the following set of equations relating the quaternion vector to the Euler angles can be used:

$$\mathbf{E}=\begin{bmatrix}\varphi\\ \theta\\ \psi\end{bmatrix}=\begin{bmatrix}\arctan\left(\dfrac{2q_0q_2-2q_1q_3}{1-2q_2^2-2q_3^2}\right)\\ \arcsin\left(2q_1q_2+2q_0q_3\right)\\ \arctan\left(\dfrac{2q_0q_1-2q_2q_3}{1-2q_1^2-2q_3^2}\right)\end{bmatrix} \tag{46}$$

where **E** denotes the set of Euler angles using the orientation presented in the previous chapter and $q_i, i=0,1,2,3$ are the elements of the quaternion vector introduced by Eq. (37). However, since the data are provided by imperfect sensors, the modeling should include the noise term. Hence, the measurement equation can be written in the form:

$$\mathbf{E}_m(t)=\mathbf{E}(t)+\mathbf{v}(t) \tag{47}$$

where $\mathbf{E}_m(t)$ is the available set of measured Euler angles and $\mathbf{v}(t)$ is a three-dimensional noise vector consisting of independent white-noise sequences.

Let the nonlinear function presented by Eq. (46) be denoted by *h*, that is $\mathbf{E}(t)=h\left[\mathbf{q}(t)\right]$. Encapsulating all the above, the state space model can be fully determined by the following set of equations:

$$\begin{bmatrix}\dot{\mathbf{q}}(t)\\ \dot{\mathbf{b}}(t)\end{bmatrix}=\begin{bmatrix}\dfrac{1}{2}\Omega\left[\boldsymbol{\omega}_m(t)-\mathbf{b}(t)-\mathbf{n}(t)\right]\mathbf{q}(t)\\ \mathbf{w}(t)\end{bmatrix} \tag{48}$$

$$\mathbf{E}_m(t)=h\left[\mathbf{q}(t)\right]+\mathbf{v}(t). \tag{49}$$

The final step is the discretization of the existing model. Assuming that the sampling interval is equal to Δt, Eqs. (48) and (49) lead to the following discrete-time state space model:

$$\begin{bmatrix}\mathbf{q}\left[k+1\right]\\ \mathbf{b}\left[k+1\right]\end{bmatrix}=\begin{bmatrix}\mathbf{q}\left[k\right]\\ \mathbf{b}\left[k\right]\end{bmatrix}+\begin{bmatrix}\dfrac{1}{2}\Omega\left[\boldsymbol{\omega}_m\left[k\right]-\mathbf{b}\left[k\right]-\mathbf{n}\left[k\right]\right]\mathbf{q}\left[k\right]\\ \mathbf{w}\left[k\right]\end{bmatrix}\Delta t \tag{50}$$

$$\mathbf{E}_m[k] = h(\mathbf{q}[k]) + \mathbf{v}[k]. \tag{51}$$

The accuracy of the model depends on the variances of the noise sequences $\mathbf{n}[k], \mathbf{w}[k]$ and $\mathbf{v}[k]$, as well as the choice of the sampling interval Δt. As indicated by Eq. (50) the value of the matrix Ω is considered to be constant between two successive iterations. This is an approximation resulting from the discretization of the model. The selection of a small value of the sampling interval is required in order to avoid the build-up of large-scale errors in the estimation process.

D. SIMPLIFIED DYNAMIC MODEL FOR THE UNBIASED ONE-AXIS ROTATION CASE

The model described above contains a 7-dimensional state vector which makes the mathematical modeling quite complex. However, when estimating an unbiased one-axis rotation the representation provided by Eqs. (50) and (51) can be significantly reduced. The assumptions for the existence of noise sequences in both angular rate and roll angle measurements still hold. These are the external disturbances which affect the attitude determination in an undesirable way and their impact is to be eliminated using the particle filter implementation.

In the one-axis rotation case described in this section, let the rotation be around the *x*-axis, so both pitch and yaw angles are equal to zero, at all times. Hence, the second and third elements of the Euler angle vector can be ignored. The same consideration can be applied to the angular rate vector, with only the *x*-axis component not being zero. As a consequence, the q_2 and q_3 elements of the quaternion vector can be ignored in Eq. (37) since no rotation around the *y* or *z* axis occurs. This leaves only two of the quaternion elements with a nonzero value, q_0 and q_1. Applying the normalization constraint introduced in Eq. (38), an implementation using a scalar state can be accomplished.

The model equations are derived using q_1 as the state of interest. The normalization constraint becomes:

$$q_0^2 + q_1^2 = 1 \Rightarrow q_1 = \pm\sqrt{1 - q_0^2}. \tag{52}$$

The 4-dimensional matrix Ω can be expressed in terms of the measured angular rate around the x-axis ω_m using the Eq. (41):

$$\Omega = \begin{bmatrix} 0 & 0 & 0 & \omega_m \\ 0 & 0 & \omega_m & 0 \\ 0 & -\omega_m & 0 & 0 \\ -\omega_m & 0 & 0 & 0 \end{bmatrix}. \tag{53}$$

Then, the attitude kinematics Equation (40) can be reduced to:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -\omega_m q_1 \\ \omega_m q_0 \end{bmatrix}. \tag{54}$$

Substitution of Eq. (52) into Eq. (54) and discretization yields:

$$q_1[k+1] = \omega_m[k]\sqrt{1-(q_1[k])^2}\,\Delta t + q_1[k]. \tag{55}$$

Similarly, the remaining Euler angle of interest (i.e., the roll angle φ), from Eq. (46), is given by:

$$\varphi[k] = \arctan\left\{ \frac{2q_1[k]\sqrt{1-(q_1[k])^2}}{1-2(q_1[k])^2} \right\}. \tag{56}$$

Let $n[k]$ and $v[k]$ be the noise sequences related to the angular rate and the roll angle measurement, respectively. The final state and measurement equations describing the one-axis rotation model can be determined as follows:

$$q_1[k+1] = (\omega[k]+n[k])\sqrt{1-(q_1[k])^2}\,\Delta t + q_1[k]. \tag{57}$$

$$\varphi[k] = \arctan\left\{ \frac{2q_1[k]\sqrt{1-(q_1[k])^2}}{1-2(q_1[k])^2} \right\} + v[k]. \tag{58}$$

In this chapter, the target system (i.e., the NPS TASS) was presented and the estimation problem was established. The dynamic model was derived and simplified for the scalar case of the one-axis rotation. The development of the PF and its evaluation in providing an adequate solution to the compensation of the additive noise is the main topic of interest discussed in the following chapter.

IV. PERFORMANCE ANALYSIS

This chapter presents the simulation results for evaluating the particle filter estimator in various cases. Firstly, KF and PF are applied to a linear model in a Gaussian environment and the estimates are compared in terms of the mean square error (MSE). Secondly, the same model in additive biased non-Gaussian noise is tested using various PF implementations. Lastly, a PF estimator for the attitude determination of the spacecraft platform discussed in the previous chapter is presented. Throughout the chapter, the importance of the right choice of the number of particles for the PF implementation is discussed.

A. PF EVALUATION FOR THE LINEAR MODEL

It is well known that KF is the optimal estimator for linear Gaussian models [2]. The reason is the fact that all *a priori* and *a posteriori* probability density functions are Gaussian and, therefore, fully defined by their mean and covariance.

It is reasonable to expect that a PF with a large number of particles provides results comparable to the KF for a linear Gaussian model. In this section the tradeoff between the number of particles (and therefore the computational cost) and the PF performance is evaluated for various examples.

The linear model that is used in this simulation section is the one described by the following set of equations:

$$x\left[k+1\right]=0.9x\left[k\right]+w\left[k\right] \tag{59}$$

$$y\left[k\right]=x\left[k\right]+v\left[k\right] \tag{60}$$

where $x\left[k\right]$ is the state to be estimated, $y\left[k\right]$ is the available measurement, $w\left[k\right]$ and $v\left[k\right]$ are white noise sequences with unit variance. The convenience that this particular measurement equation provides is the direct association of the state to the measurement, so the measurement error can be directly compared to the state error provided by both filters. The initial state value was assumed to be equal to zero in all simulations.

A simulation sample result regarding the performance of the Kalman filter for this model is presented in Fig. 4. In this simulation a sequence of 100 points was created based on Eq. (59). Then the Eq. (60) was used to provide an available set of measurements. The Kalman filter algorithm was implemented and the filter output is plotted along with the absolute error of the estimation. It is clear that the filtered output provides much better estimation of the state than the available set of measurements. A measure to evaluate the performance of the Kalman filter is the mean square error (MSE). The MSE of the estimation was computed to be 0.44857 while the MSE of the measurement was 0.98838. This means that in this case the use of KF decreases the MSE of the estimation by 54.6%.

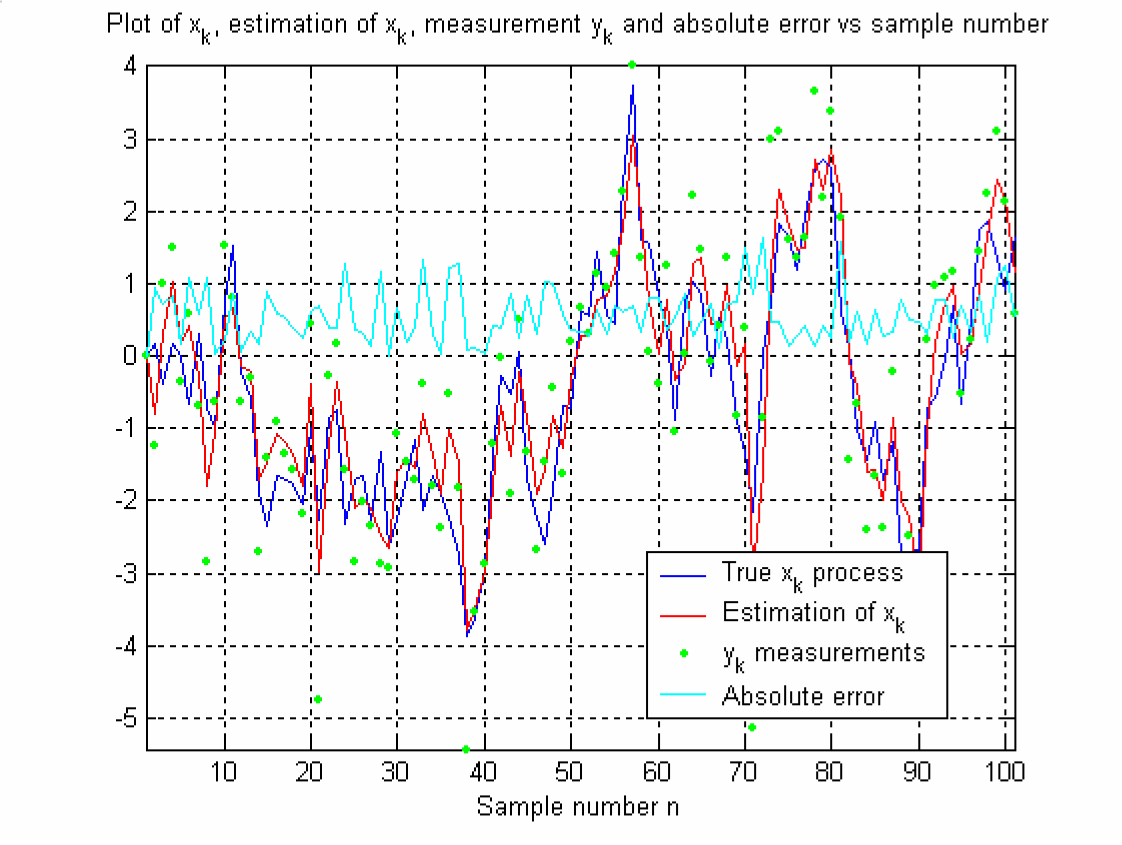


Figure 4. Demonstration of KF performance

In order to compare the KF with the PF estimates, numerous simulations were executed for different number of particles for the PF. In Figs. 5 through 10 the results of 1000 simulations per case are presented in the form of histograms. In particular, the results testing the PF for 10, 20, 100, 500, 2000 and 5000 particles are analyzed. For each simulation, a sequence of 50 points (i.e., time steps), based on the model described by Eqs. (59) and (60), was generated and estimated using both Kalman and particle filters. The measure of comparison between the Kalman filter and the PF is the MSE.

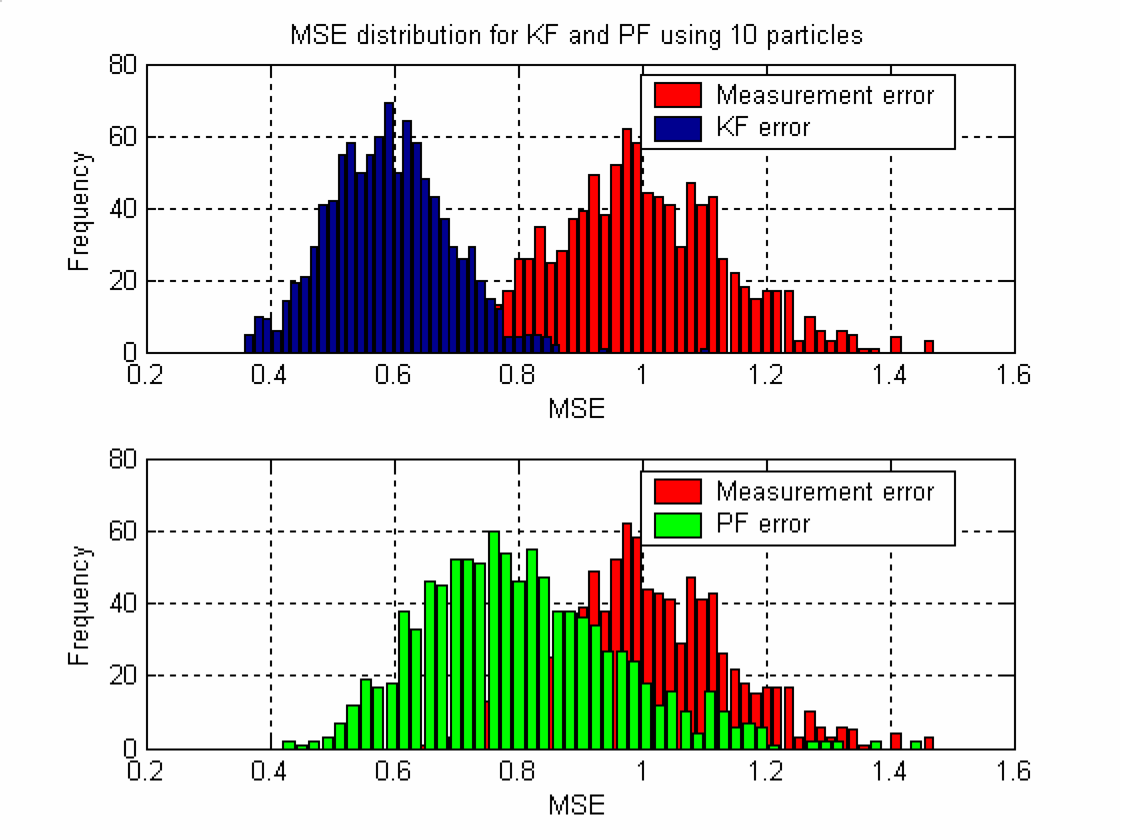


Figure 5. MSE comparison for KF and PF using 10 particles

In the first case, 10 particles were used for the PF implementation. For each simulation the MSE of the measurement, the KF and the PF were computed and plotted as single points. The results are shown in Fig. 5. It is clear that in this case the KF outperforms the PF. The percentage of better KF performance was computed to be 99.2% of the tested runs. Finally, the average MSEs of all simulations were computed to be 0.99848 for the measurement, 0.59178 for the KF estimate and 0.80476 for the PF.

The implication of these results is that even if the PF reduces the measurement error, it is unlikely to provide a good estimate for the state. This is due to the insufficient number of particles used in this simulation. The next simulation where 20 particles are used indicates the improvement of the PF performance when more particles are used. These results are presented in Fig. 6.

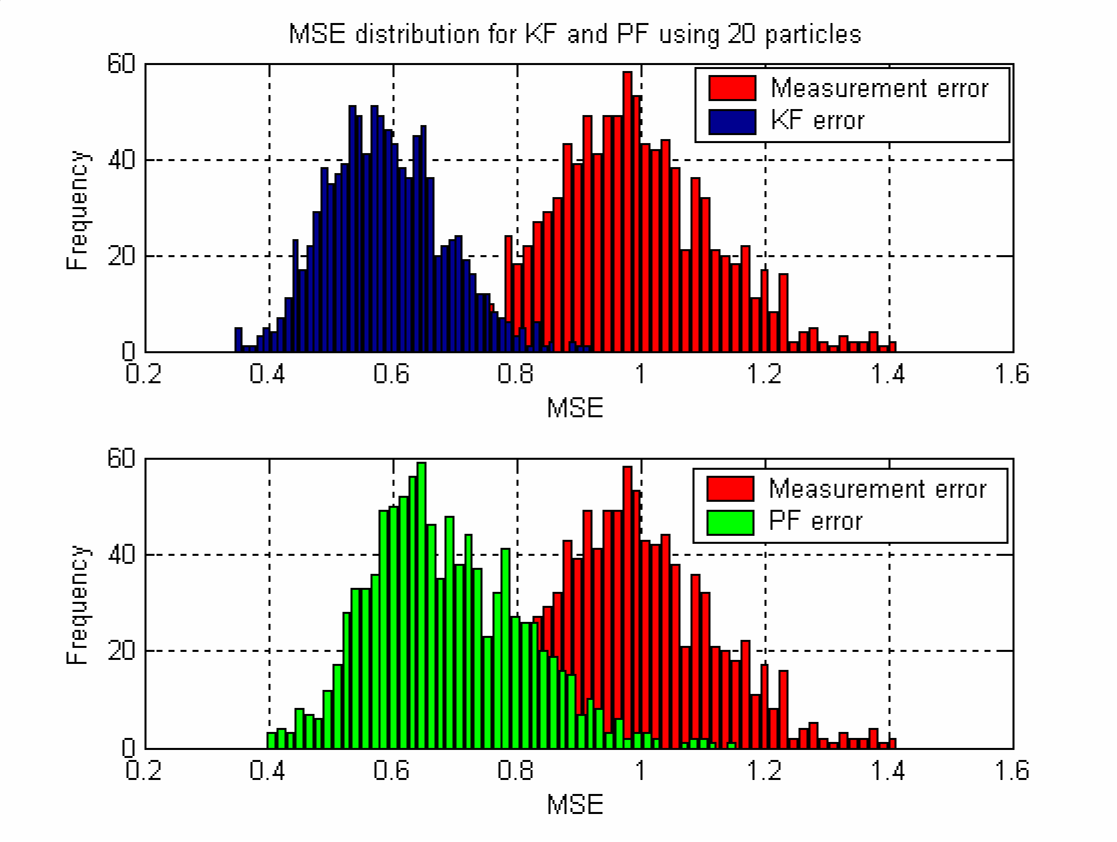


Figure 6. MSE comparison for KF and PF using 20 particles

Comparing Figs. 5 and 6, the improvement of the PF is obvious. The KF still outperforms PF by a significant percentage of 94.6%. This implies a still poor performance for the PF. The average MSEs of all simulations were computed to be 0.98340 for the measurement, 0.58888 for the KF estimate and 0.68623 for the PF. Obviously, there is more improvement to be accomplished by selecting more particles to be propagated by the PF.

The next set of simulations was executed by choosing 100 particles for the parameter of interest. The resulting plot is shown in Fig. 7. The increase of the number of particles to this value has resulted in a significant improvement of the PF performance. The average MSEs of all simulations were computed to be 0.99105 for the measurement, 0.59027 for the KF estimate and 0.61723 for the PF one. The improvement of the PF performance can be easily visualized by the reduction of the distance between the plotted KF and PF MSEs that can be observed in Fig. 7 when compared to Figs. 5 and 6.

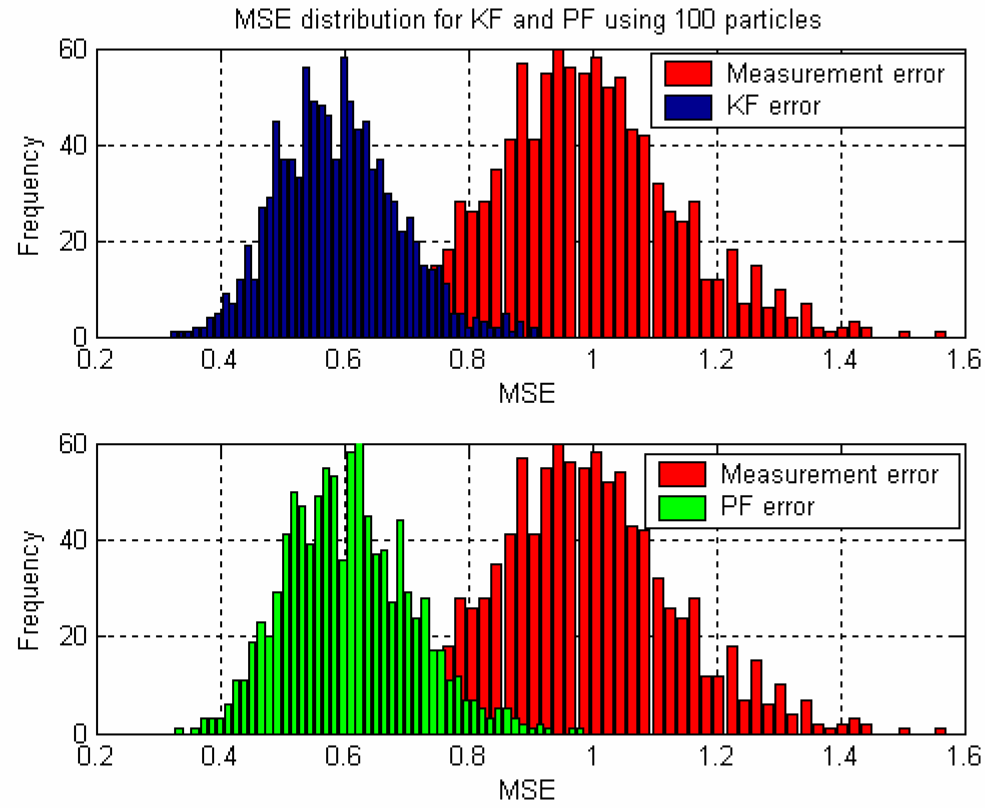


Figure 7. MSE comparison for KF and PF using 100 particles

Figure 8 presents the results using 500 particles. The PF performance is once more improved. The average MSEs of all simulations are 0.97850 for the measurement, 0.58419 for the KF estimate and 0.58812 for the PF one. The percentage of better KF performance is reduced to 62.9%. Recall that the PF converges to the KF one in the theoretical case of infinite number of particles. This means that from a statistical point of view the lower bound for better KF performance is 50%. Evaluating the PF, the conclusion is that 500 can establish a lower bound for the number of particles in order to get results of high accuracy for this estimation problem.

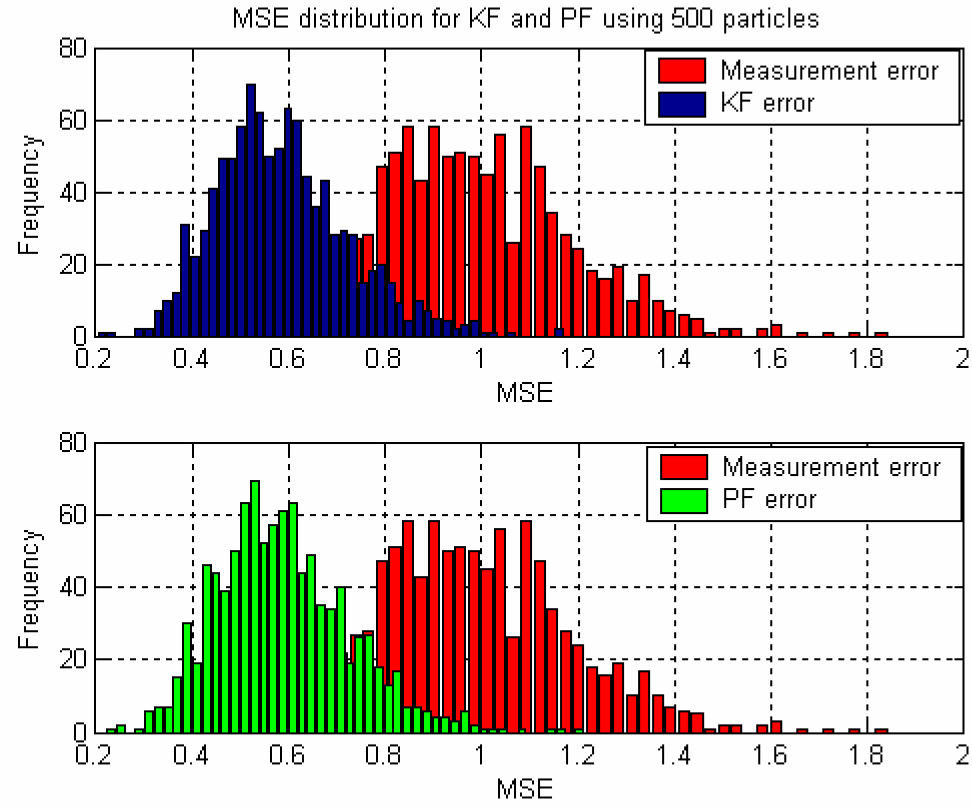


Figure 8. MSE comparison for KF and PF using 500 particles

Concluding the research for the linear model case, two more sets of simulations using 2000 and 5000 particles were executed. The resulting plots are shown in Figs. 9 and 10, respectively. For the 2000 particles case, the average MSEs of all simulations are 0.98517 for the measurement, 0.58908 for the KF estimate and 0.58977 for the PF one. The percentage of better KF performance is computed to be 62.6%. The performance of the PF is evaluated as sufficient, considering that the average distance ratio between the MSEs of the KF and the PF is less than 0.12%. This fact can be easily visualized in the related figure by noticing that for a significant number of simulations the green bars representing the PF MSE lay partially over the blue bars representing the KF MSE.

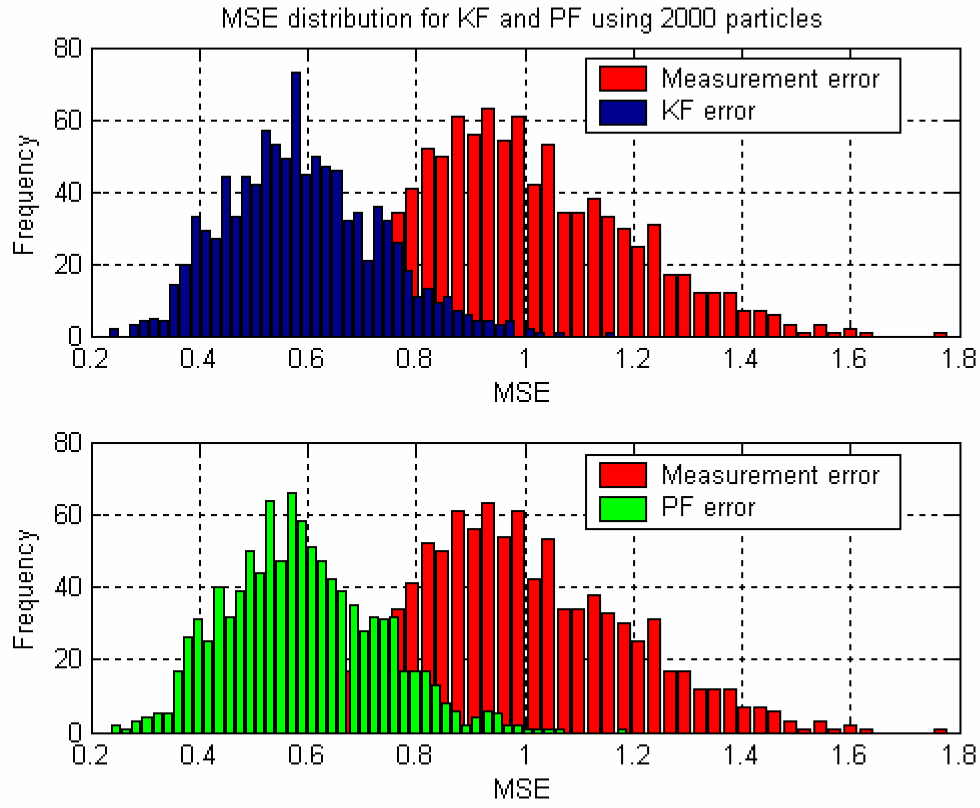


Figure 9. MSE comparison for KF and PF using 2000 particles

The resulting plot for the 5000 particles case is shown in Fig. 10. The percentage of better KF performance is 66.1% indicating that the required number of particles for the particular application has exceeded its saturation point. The resulting average MSEs are 0.98403 for the measurement, 0.58900 for the KF estimate and 0.58945 for the PF one. The performance of PF can be evaluated as excellent. However, the computational effort required to provide these results is considerable. The average distance ratio between the MSEs of the KF and the PF is less than 0.08%, verifying the tradeoff between the computational cost and the provided estimation accuracy.

For further reference, all simulation results are also presented in Appendix A in the form of scatter plots.

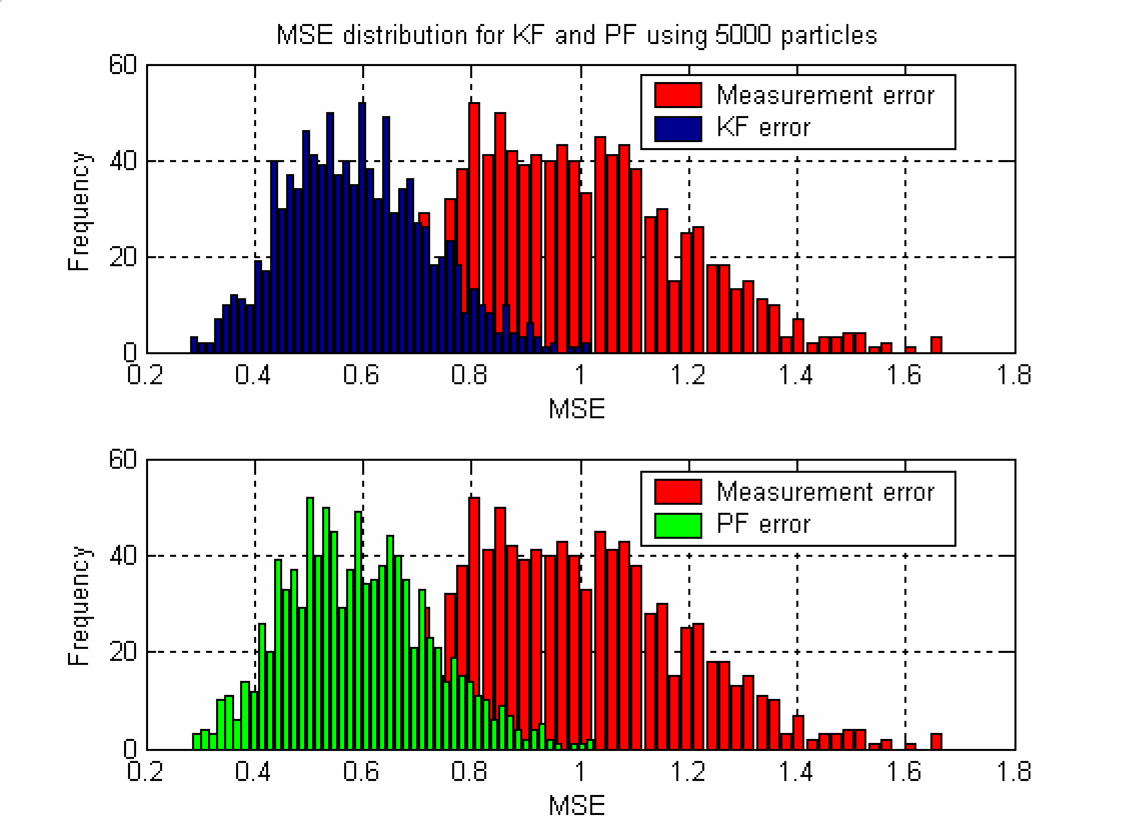


Figure 10. MSE comparison for KF and PF using 5000 particles

The results of all simulation sets are summarized and tabulated in Table 1. The improvement ratio for each set is computed with respect to the measurement error, which is defined as:

$$I_{KF} = \frac{\sum_{i=1}^{1000} MSE_{meas}^{i}}{\sum_{i=1}^{1000} MSE_{KF}^{i}} \tag{61}$$

$$I_{PF} = \frac{\sum_{i=1}^{1000} MSE_{meas}^{i}}{\sum_{i=1}^{1000} MSE_{PF}^{i}} \tag{62}$$

where I stands for the improvement ratio, subscripts KF and PF represent the related filter estimates, subscript $meas$ refers to the measurement and i represents the simulation index number for each set.

Sim set #	Number of particles used	Average MSE_{meas}	Average MSE_{KF}	Average MSE_{PF}	I_{KF}	I_{PF}
1	10	0.99848	0.59178	0.80476	1.68725	1.24718
2	20	0.98340	0.58888	0.68623	1.66995	1.43305
3	100	0.99105	0.59027	0.61723	1.67898	1.60564
4	500	0.97850	0.58419	0.58812	1.67497	1.66378
5	2000	0.98517	0.58908	0.58977	1.67239	1.67043
6	5000	0.98403	0.58900	0.58945	1.67068	1.66940

Table 1. MSE comparison chart of KF and PF performance for the linear model case

To complete the study for this section, several conclusions can be mentioned. First, the number of particles to be propagated through the PF is important to the filter performance. Second, for real-time applications the computational intensity is an important issue. Hence, the selection of the amount of particles has to be made with respect to both accuracy and complexity. As a guideline, for the particular model tested, a selection of 500 to 2000 particles would be a wise choice.

B. PF EVALUATION IN NON-GAUSSIAN ENVIRONMENT

The purpose of the simulations of this section is to demonstrate the robustness of PF in any kind of additive noise. For this purpose the model described by Eqs. (59) and (60) is used to generate the state and measurement sequences. The only assumption different than the ones made in the previous section is that the process noise sequence $w[k]$ is the square of a white Gaussian noise of unit variance and, therefore, non-Gaussian with mean value equal to 1 and standard deviation equal to $\sqrt{2}$.

The KF is not expected to work effectively in this case, so it was not included in the resulting plots. The comparison can be made between the MSE of the measurement and the MSE of the estimation, since the state and the measurement are directly associated with no imposed scaling or addition factor through the measurement equation. For each simulation, sequences of 100 points (i.e., time steps) were generated. Each simulation set consists of 1000 runs. The number of particles is the parameter of interest and the PF was tested for 10, 50, 100, 500 and 2000 propagating particles. The results are presented in Figs. 11 through 15.

The resulting plot for the 10 particles case is shown in Fig. 11. The impact of the non-Gaussian environment is quite obvious. In most cases the PF is unable to provide better estimation than the measurement itself. This is due to the increased value of the process noise variance plus the biased conditions for the state equation. The bad performance of the PF can be outlined by the average MSEs which are 0.99083 for the measurement and 1.4976 for the estimate. An increase of the number of particles is needed in order to reach filter convergence.

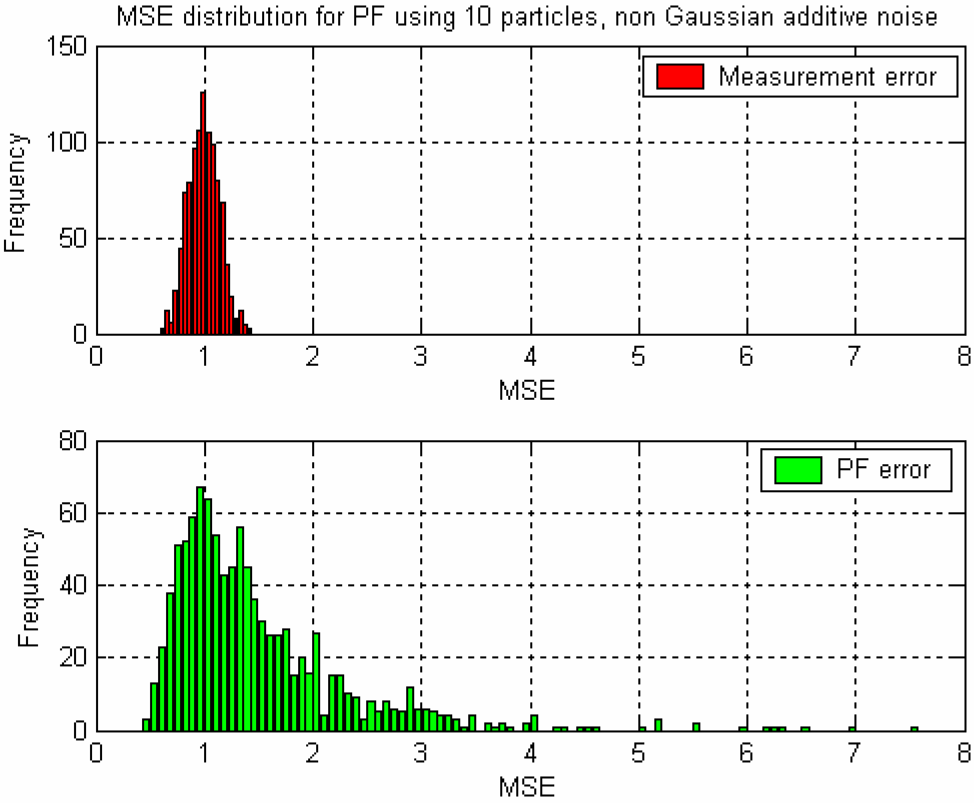


Figure 11. MSEs for the non-Gaussian noise case using 10 particles

The next simulation was executed using 50 particles. It is not expected to provide acceptable performance, but the results can be used for comparison. The resulting plot is presented in Fig. 12.

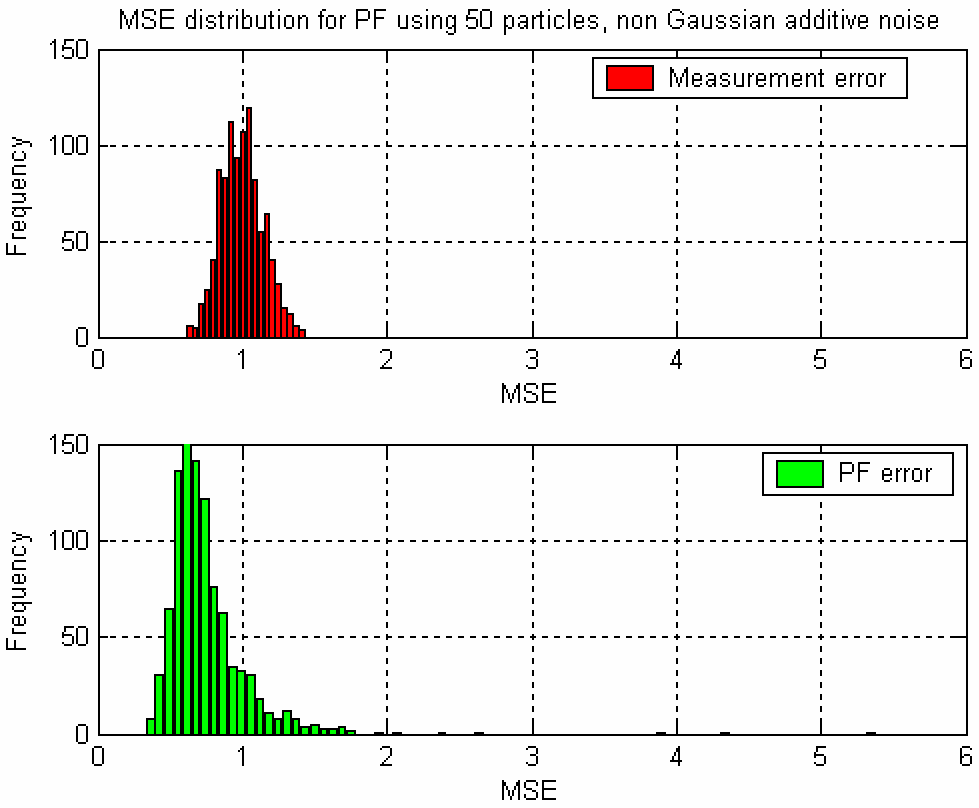


Figure 12. MSEs for the non-Gaussian noise case using 50 particles

The significant improvement of the PF performance is obvious when comparing Figs. 11 and 12. For the 50-particle case, the estimates are better than the measurements themselves in most cases, indicating filter convergence. The average MSEs were computed to be 0.99605 for the measurement and 0.75424 for the PF estimate.

From the results presented to this point, the conclusion made in the linear case on the selection of the number of particles would be expected to hold in this case as well. Hence, the simulations for 500 and 2000 particles are of significant importance. For comparison reasons, the simulation for 100 particles was executed and the resulting plot is shown in Fig. 13. The resulting plots for the other two cases (500 and 2000 particles) are shown in Figs. 14 and 15, respectively.

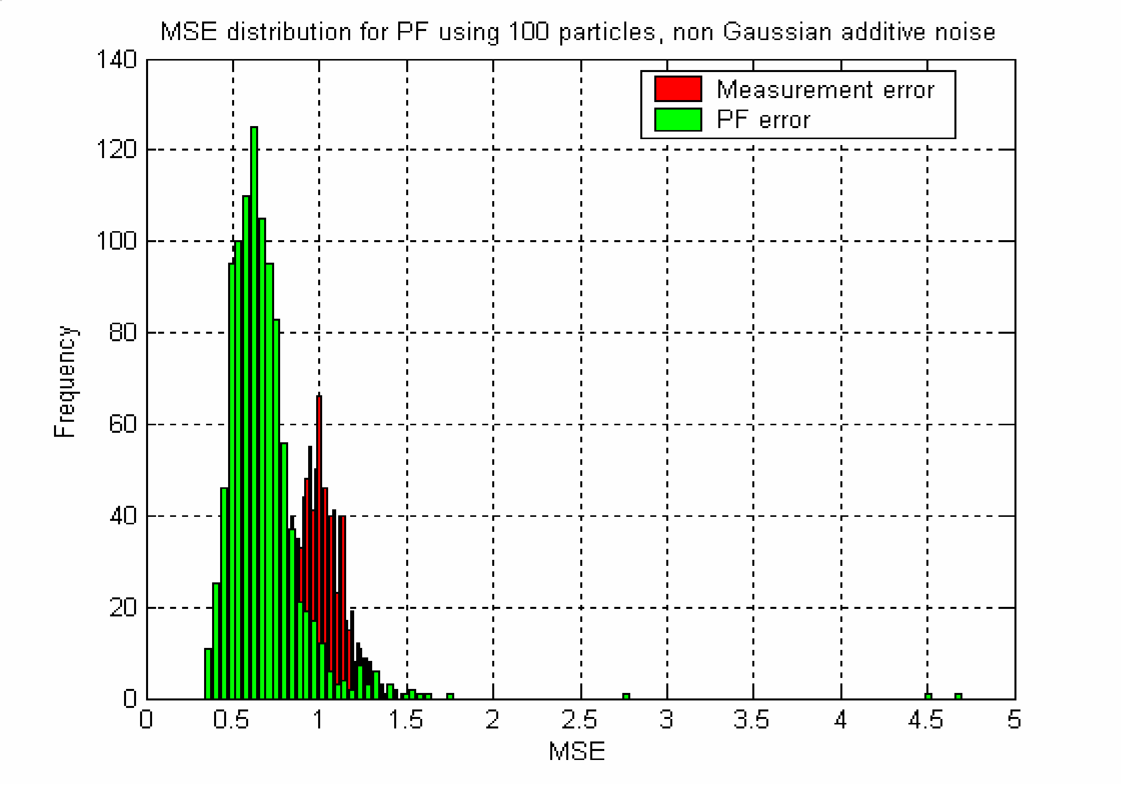


Figure 13. MSEs for the non-Gaussian noise case using 100 particles

The results of Fig. 13 indicate extensive improvement for the PF. The average MSEs were computed to be 0.99227 for the measurement and 0.67816 for the PF estimate. However, in a few simulations the PF generated large errors. This fact implies the necessity to hold to the initial guideline for using 500 to 2000 particles in order to guarantee estimates better than the measurements.

Obviously, both implementations for 500 and 2000 particles seem to provide good estimates of the state. The improved performance of the PF is indicated by the average MSEs as well. All of the computed values for the average MSEs for both Gaussian and biased non-Gaussian noise cases regarding the PF are summarized and tabulated in Table 2. The same table contains the computed improvement ratios for the PF as presented in Eq. (62) in order to provide a general measure of comparison for all simulated implementations.

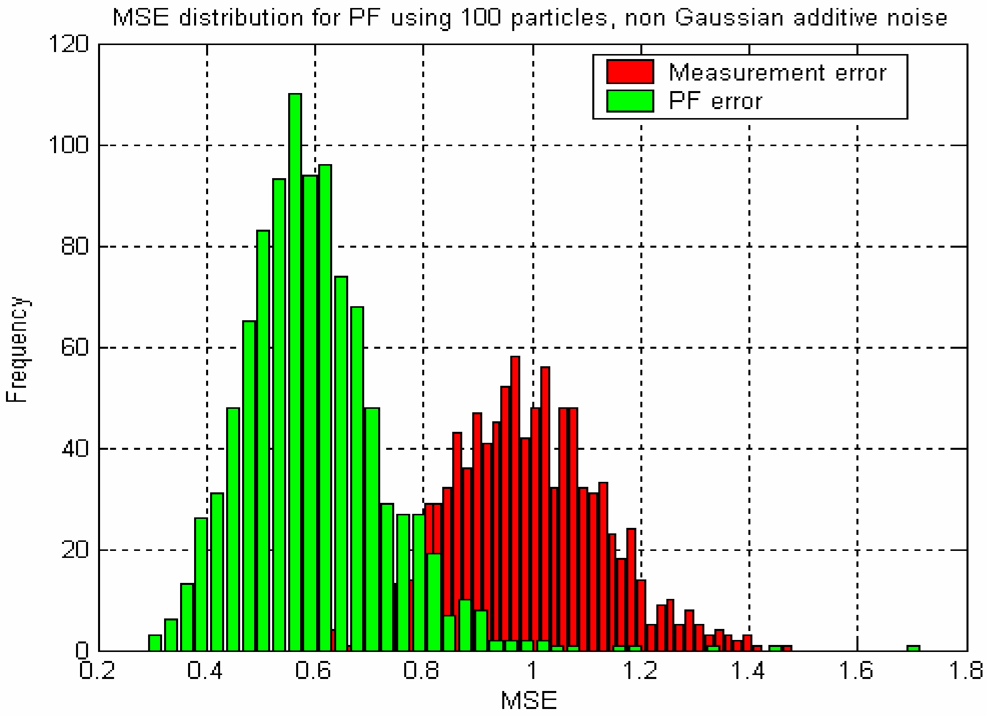


Figure 14. MSEs for the non-Gaussian noise case using 500 particles

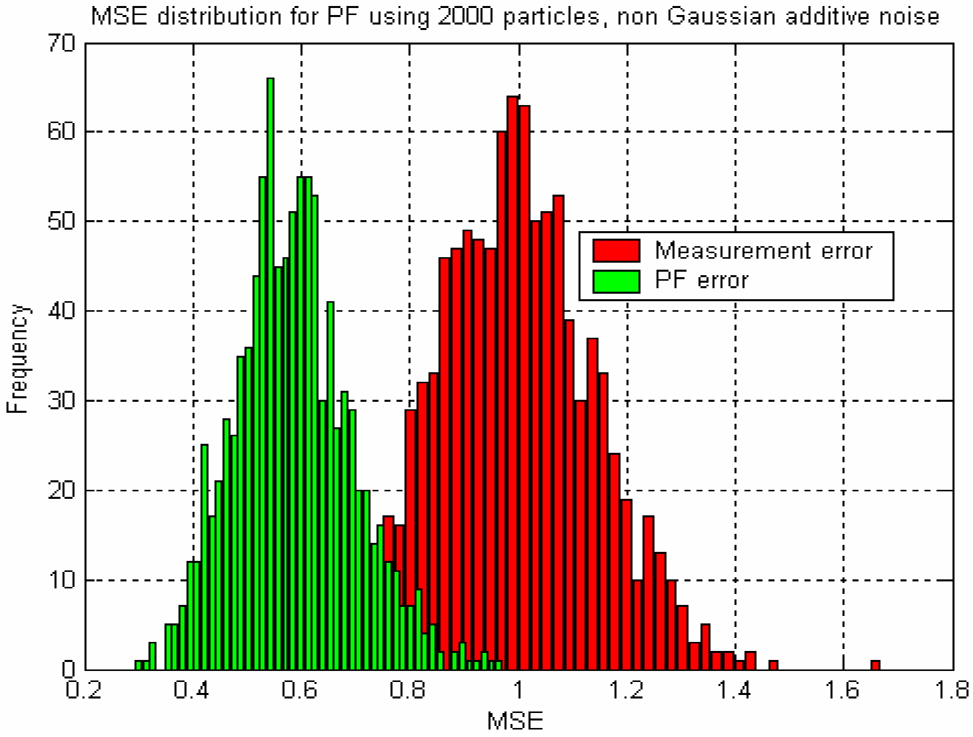


Figure 15. MSEs for the non Gaussian noise case using 2000 particles

Number of particles used	Linear model in Gaussian noise			Linear model in non Gaussian noise		
	Average MSE_{meas}	Average MSE_{PF}	I_{PF}	Average MSE_{meas}	Average MSE_{PF}	I_{PF}
10	0.99848	0.80476	1.24718	0.99083	1.49760	0.66161
20	0.98340	0.68623	1.43305	-	-	-
50	-	-	-	0.99605	0.75424	1.32060
100	0.99105	0.61723	1.60564	0.99227	0.67816	1.46318
500	0.97850	0.58812	1.66378	0.98803	0.59702	1.65494
2000	0.98517	0.58977	1.67043	0.98525	0.58488	1.68453
5000	0.98403	0.58945	1.66940	-	-	-

Table 2. MSE comparison chart for PF performance in linear model implementations

The results for the two linear model cases show that the nature of the additive noise significantly affects the performance of the PF when an insufficient number of particles is used. On the contrary, when enough particles are propagated in each iteration, the PF yields robust performance regardless of the nature of the noise.

The simulation results obtained up to this point can be used to decide on the number of particles to be used in the attitude determination problem. In the literature [7] it is stated that a PF using 2000 particles would guarantee convergence for this specific application. This statement is in agreement to the conclusions that have been derived up to this point.

C. PF EVALUATION IN ATTITUDE DETERMINATION FOR THE UNBIASED ONE-AXIS ROTATION CASE

In this section, the SIR algorithm developed in Chapter II is tested for attitude determination based on the model described by the set of Eqs. (57) and (58). The simulation lasts for 20 seconds using a discretization time interval of 0.01 seconds resulting to a total of 2000 time samples, assuming zero initial conditions. The disturbance on the system is added in the form of noise in the angular rate measurements. Despite the use of the quaternion representation for the system modeling, the final evaluation is based on the error reduction in the roll angle using the quaternion estimation, due to the fact that, in practice, the Euler angle estimation is of interest.

1. Data Generation

The data are generated assuming a sinusoidal angular rate disturbance around the *x*-axis of the form:

$$\omega(t) = A\sin\left(\omega_0 t\right) \tag{63}$$

where $\omega(t)$ is the angular rate value at time *t*, and the parameters *A* and ω_0 are the amplitude and the natural frequency, respectively. Figure 16 shows the generated true values of the angular rate through time.

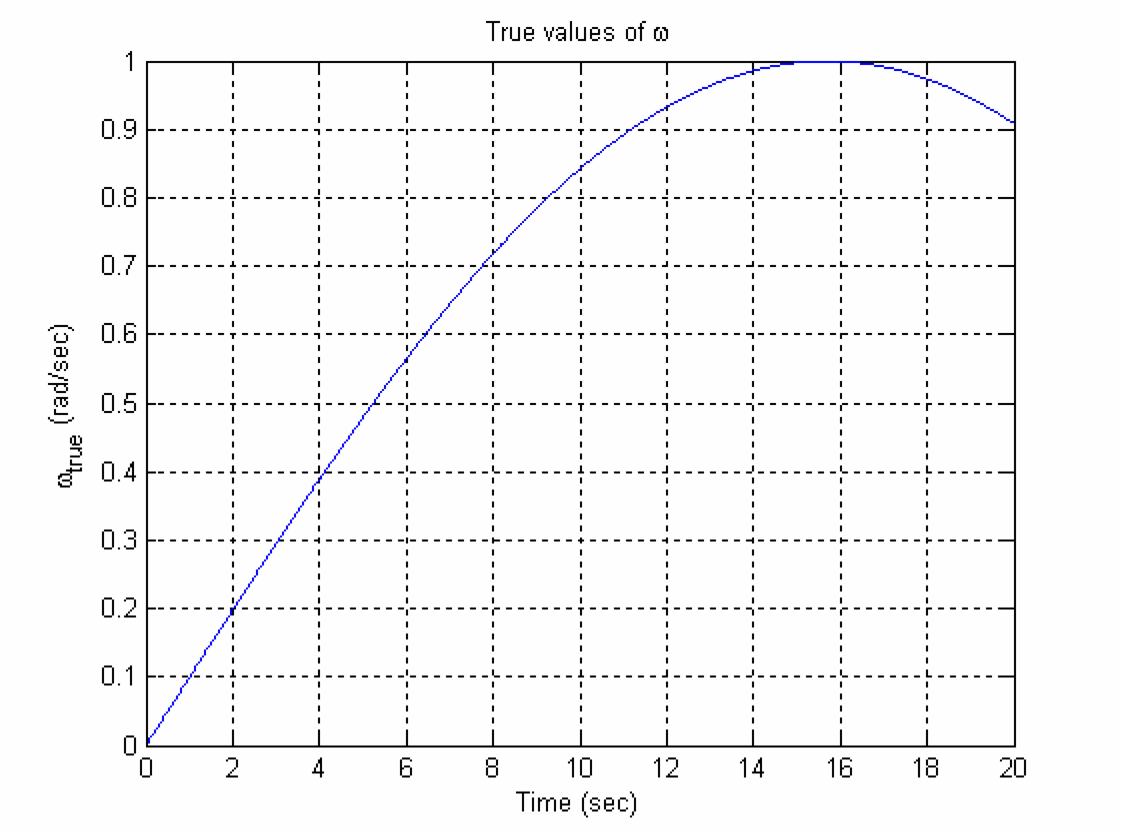


Figure 16. True values of angular rate

The parameters A and ω_0 were selected in such a way that the resulting roll angle representation included discontinuities during the simulation interval. The specific values used were $A=1$ rad/sec and $\omega_0=0.1$ rad/sec.

The noise sequences $n[k]$ and $v[k]$ presented in the system model were assumed to be white Gaussian noise with standard deviations of $\sigma_n=10^{-1}$ rad/sec and $\sigma_v=10$ degrees respectively. These values were selected in order to clearly visualize the additive noise in the associated plots. The resulting true and measured roll angles sequences are shown in Fig. 17.

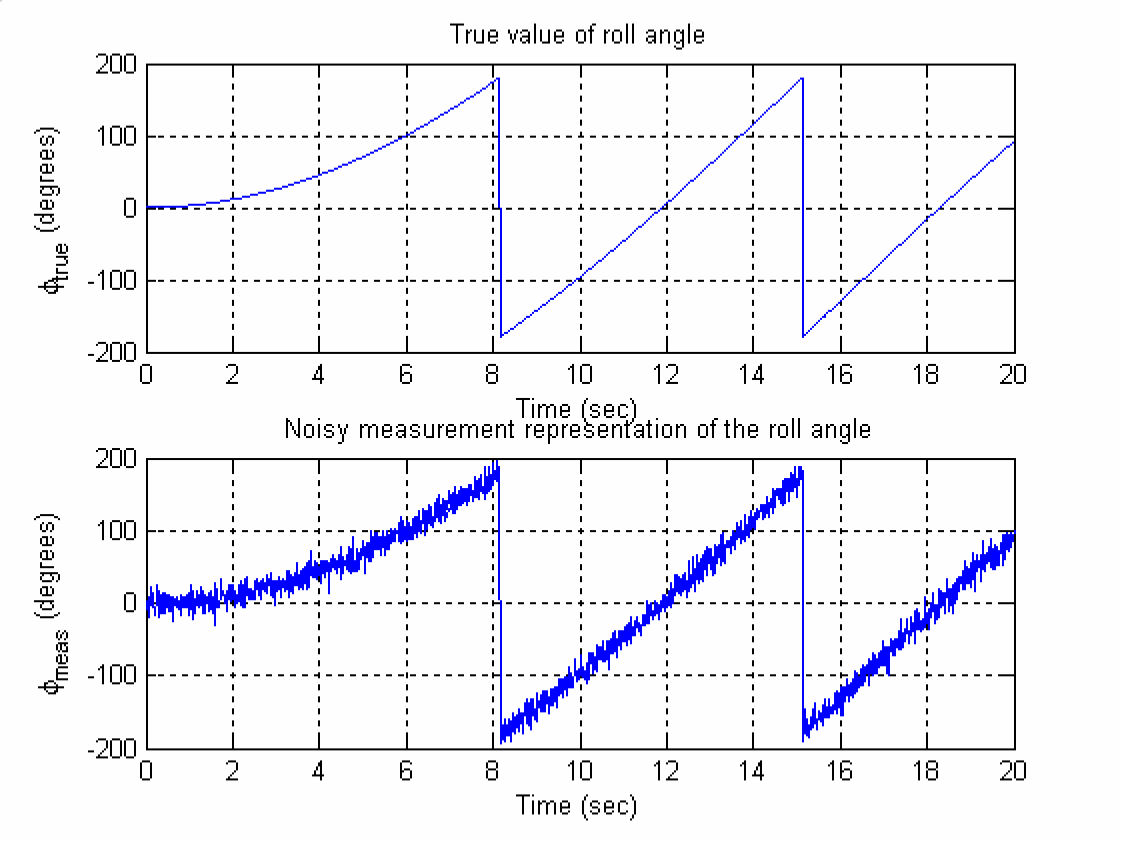


Figure 17. True and measured values of the roll angle

It is obvious that there are two discontinuities in the roll angle sequence that have to be overcome by the estimator. Due to these discontinuities, an implementation using the EKF can not provide satisfactory results and the PF is a better approach.

2. Attitude Representation

In order to proceed to the estimation step, the required quantities have to be computed. First of all, the roll angle measurement set is converted to the associated normalized quaternion set. The actual quaternion values and the ones calculated by the noisy measurement set are shown in Fig. 18.

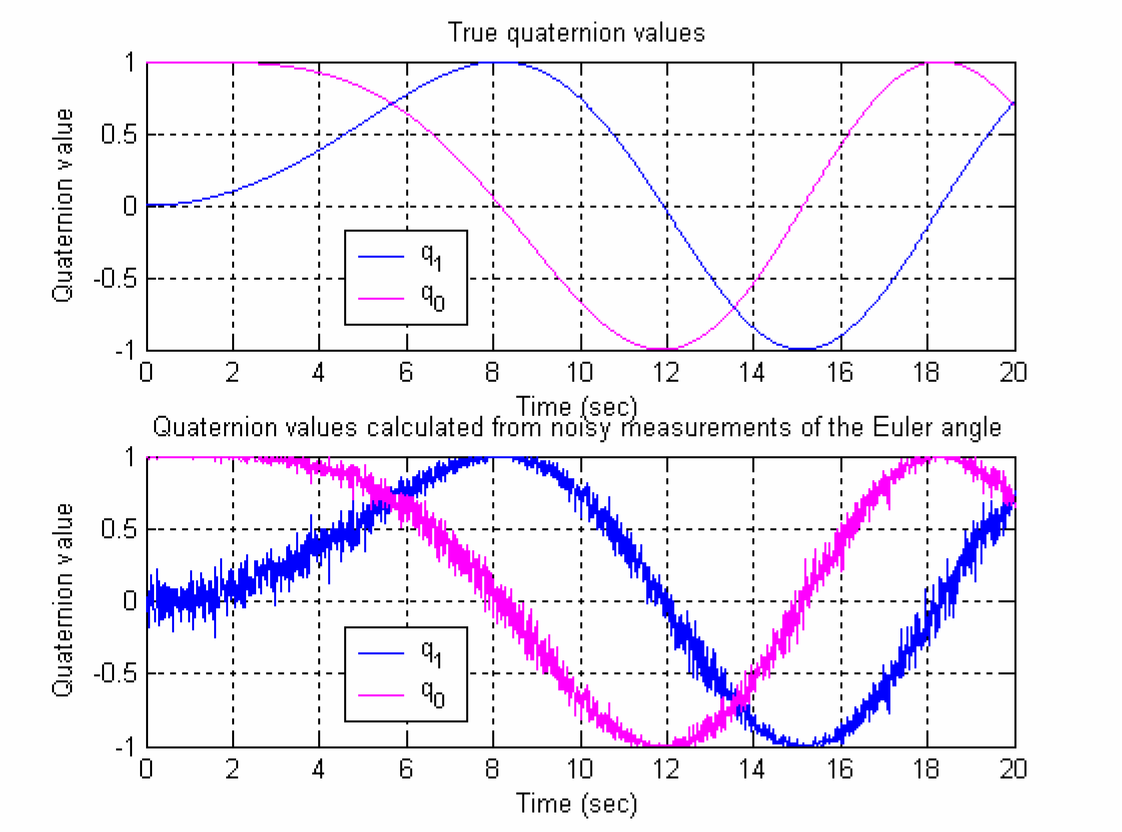


Figure 18. True and calculated from noisy measurements quaternion values

Since the dynamic model derived in the previous chapter is expressed in terms of the quaternion component q_1, this is the state to be estimated. The task to be accomplished is the minimization of the state error which is presented in Fig. 19. It is noticeable that the state error is reduced around the time indices that the quaternion receives its boundary values ± 1. This is due to the normalization constraint enforcement that prevents the state from receiving invalid values. The figure also contains the measurement error.

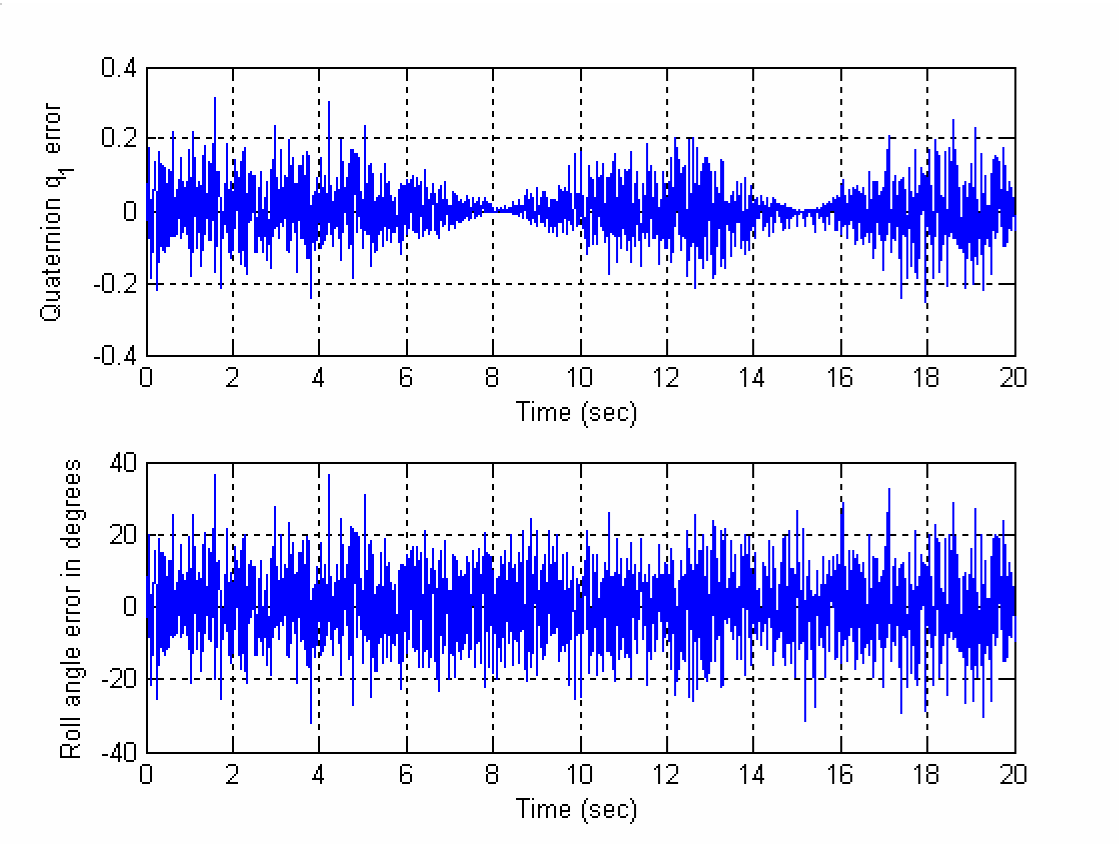


Figure 19. Quaternion and Euler angle error

3. Estimation Analysis

For the state estimation the PF approach and the dynamic model derived in Chapter III were used. The generated noisy data set for the roll angle was given as an input to the estimator. Two different implementations using 500 and 2000 particles were tested. No initial estimation error was assumed for any of the simulations and the performance was compared in terms of the MMSE criterion. The resulting estimate $\hat{q}_1$, of the quaternion component q_1, was converted to the associated roll angle generating the final estimate $\hat{\varphi}$. The estimation error, $e_{est} = \|\varphi_{true} - \hat{\varphi}\|$, was compared to the measurement error $e_{meas} = \|\varphi_{true} - \varphi_{meas}\|$.

The simulation indicated that the developed PF is capable of reducing the noise to an acceptable level. The results are presented in the following sections.

a. Estimation using 500 particles

The resulting curve of the estimated state is presented in Fig. 20. Two random parts of the figure have been scaled-up to provide better visualization of the estimation error. Obviously, there has been a significant reduction of the state error and the estimation algorithm is shown to be effective.

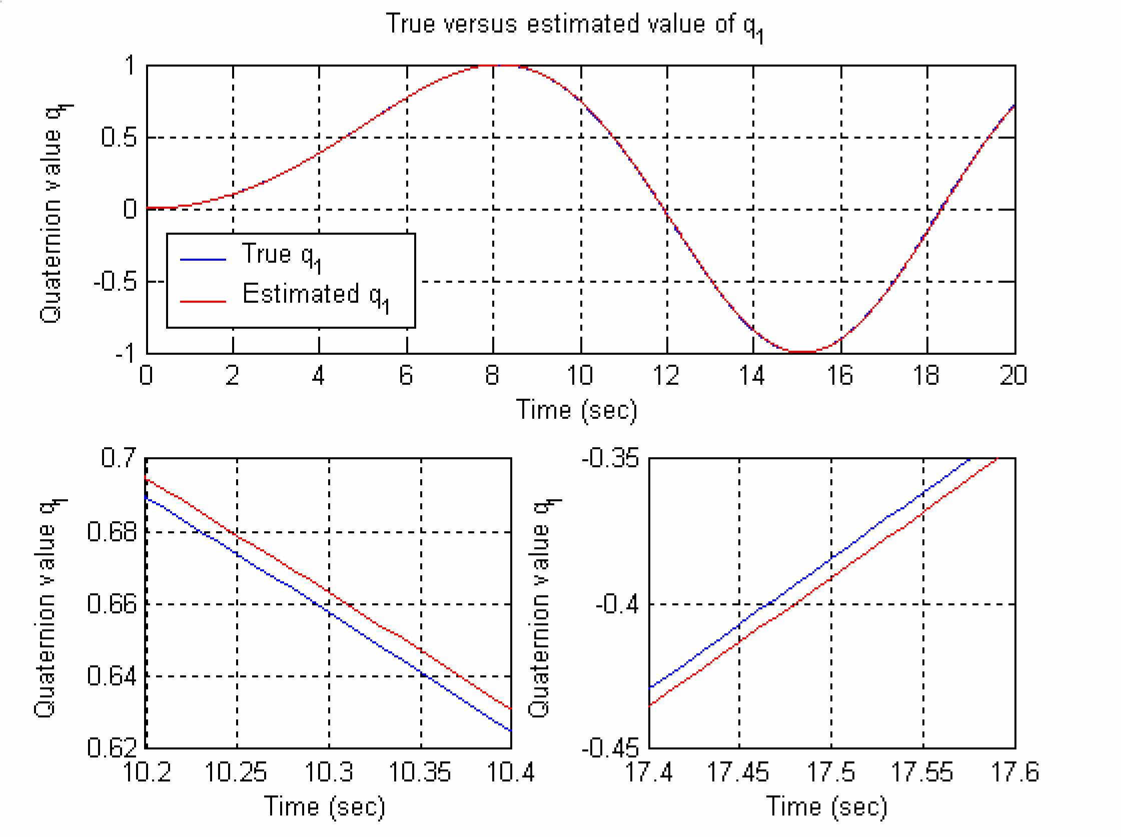


Figure 20. State estimation using 500 particles

The most crucial point for the filter evaluation is the analysis of the estimated roll angle curve which is presented in Fig. 21. The estimated curve is smooth and approximately overlaid over the actual one indicating small estimation error. Two random parts of the figure have been scaled-up to provide better visualization. The evaluation of the PF performance is done using the MSE criterion. The MSE for the available set of measurements was $(10.0255\,\deg)^2$ while the MSE of the filtered estimate was $(0.64283\,\deg)^2$. This means that, in terms of MSE, the PF provided an improvement in the error reduction by a factor of 243.23. Finally, both measurement and estimation absolute errors are presented in Fig. 22 in semi-logarithmic form.

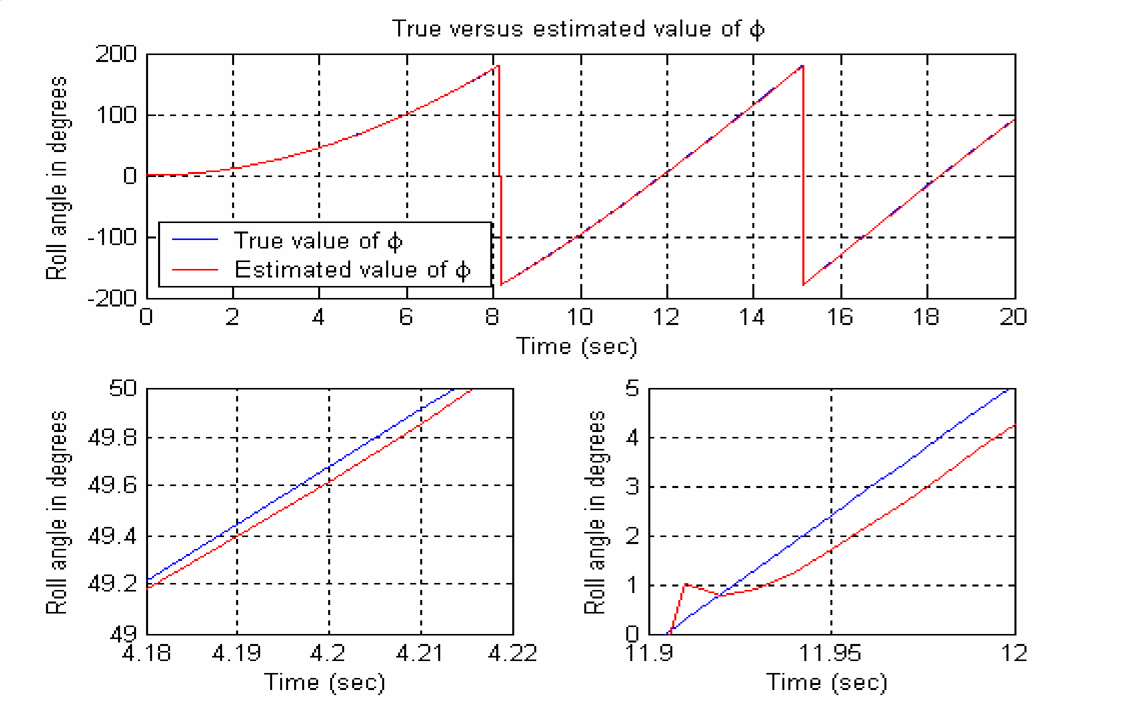


Figure 21. Roll angle estimation using 500 particles

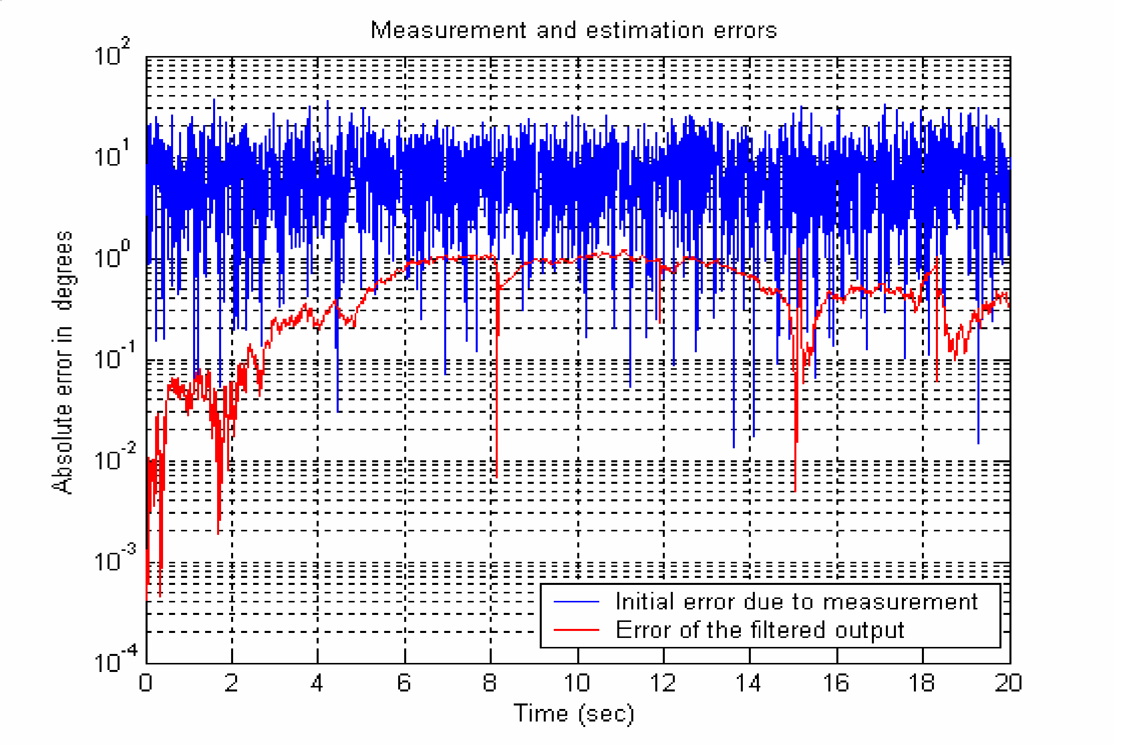


Figure 22. Comparison of the measurement and estimation errors using 500 particles

b. Estimation Using 2000 Particles

For this simulation the same set of noisy measurements was used. The increased number of particles provided even better performance. The same set of plots was generated and the Figs. 23, 24 and 25 show the resulting state estimation, the roll angle estimation and the absolute error representations, respectively.

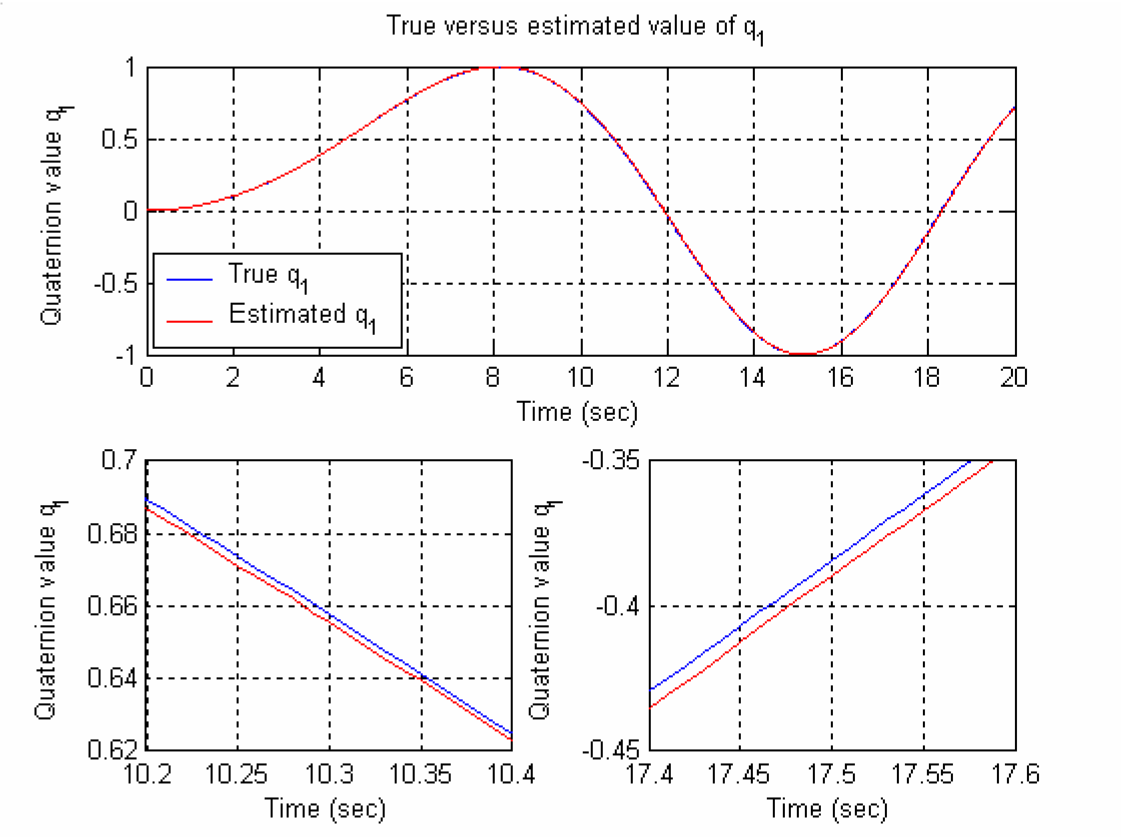


Figure 23. State estimation using 2000 particles

As expected, the state error is reduced in comparison with the 500 particles case. As a result, the roll angle estimation error is also reduced as it can be clearly noticed by Figs. 24 and 25. The resulting MSE for the roll angle estimate in this case is $(0.40982\,\deg)^2$, resulting in a factor of approximately 598 concerning the error reduction rate in terms of the MSE. Finally, all the results for both implementations are summarized and tabulated in Table 3.

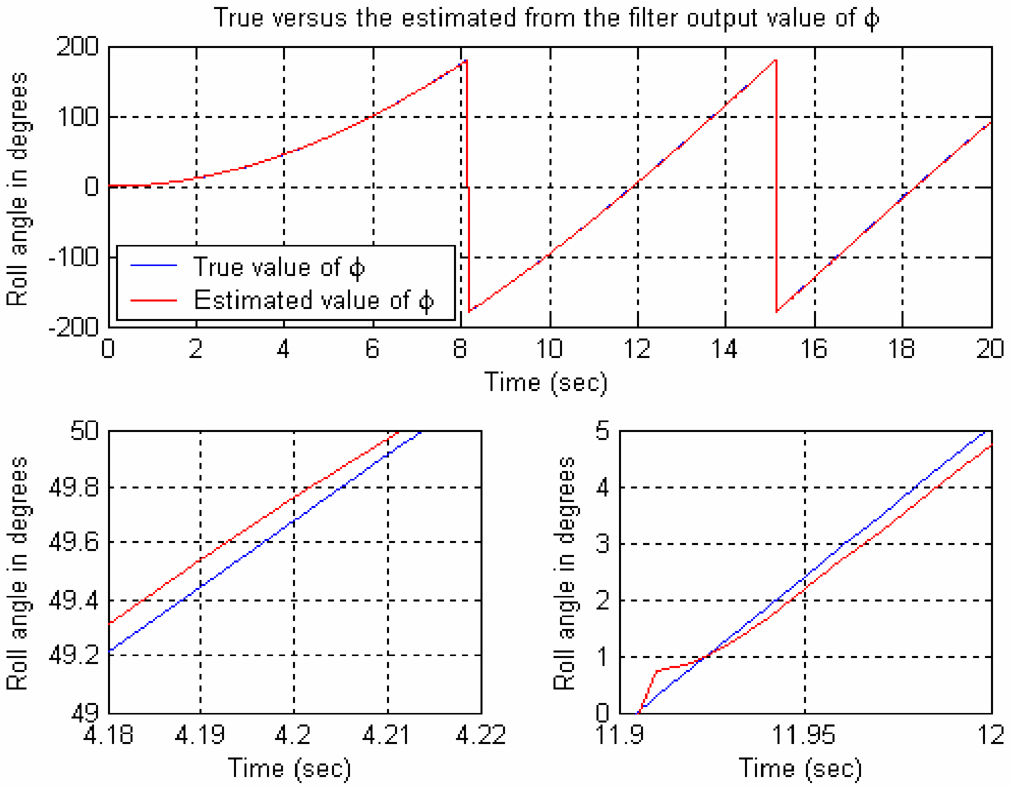


Figure 24. Roll angle estimation using 2000 particles

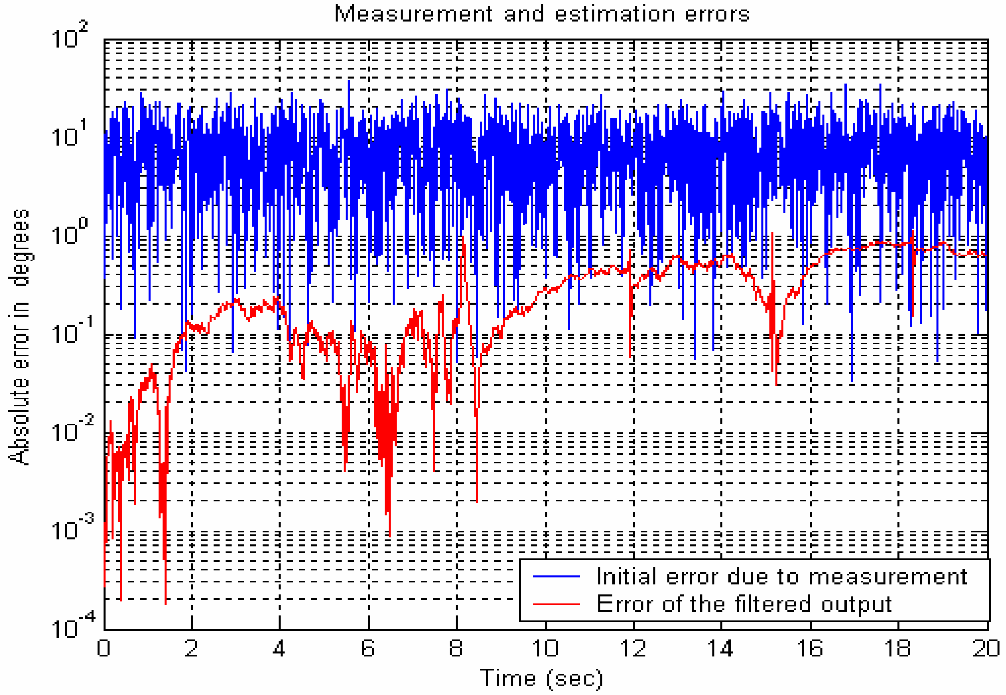


Figure 25. Comparison of the measurement and estimation errors using 2000 particles

Number of particles used	e_{meas} (deg)	MSE_{meas} (deg^2)	e_{est} (deg)	MSE_{est} (deg^2)	MSE_{meas}/MSE_{est}
500	8.0533	100.51	0.60591	0.41323	243.23
2000	8.0533	100.51	0.31663	0.16795	598.45

Table 3. MSE comparison chart for PF performance in the nonlinear application

D. SUMMARY

In this chapter, the PF (based on the developed SIR algorithm) was tested on various cases. Particular emphasis was given to the nonlinear application of attitude determination for the NPS TASS for the unbiased one-axis rotation case.

At first, the good performance of the PF was verified for the linear model regardless the statistics of the disturbances. The major impact of the number of propagated particles in determining the accuracy of the estimation was underlined. On the other hand, the computational intensity is a constraint that has to be taken under consideration with respect to any particular application.

Finally, the PF demonstrated excellent performance in the nonlinear model estimation problem. Both implementations that were tested provided high accuracy results.

The following chapter of this thesis provides the conclusions and suggests directions for future expansion of the research.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS

The PF based on the SIR algorithm can be utilized as an accurate estimator of spacecraft attitude using quaternion representation. It demonstrates robust performance even when implemented in a nonlinear dynamic model overcoming the difficulties introduced by discontinuities.

A. SUMMARY

A PF estimator for attitude angular determination has been presented in this thesis. Unlike standard implementations using the KF, the approach presented is completely general and can be adapted to any nonlinear, non-Gaussian statistical model, without the need of any linearization. In particular, the implementation to one-axis rotation has shown the effectiveness even in the presence of discontinuities in the model.

B. RECOMMENDATIONS

It is recommended that future work include the following expansions of this research:

1. Expansion to the Biased Three-axis Rotation Case

The dynamic model for spacecraft attitude determination for the biased three-axis rotation scenario has been derived in this thesis. The PF implementation has to be expanded to include the 7-dimensional state vector and some simulations have to be executed. This research will provide much more realistic results regarding the final utilization of PF in this specific application.

2. Gibbs Vector Parameterization

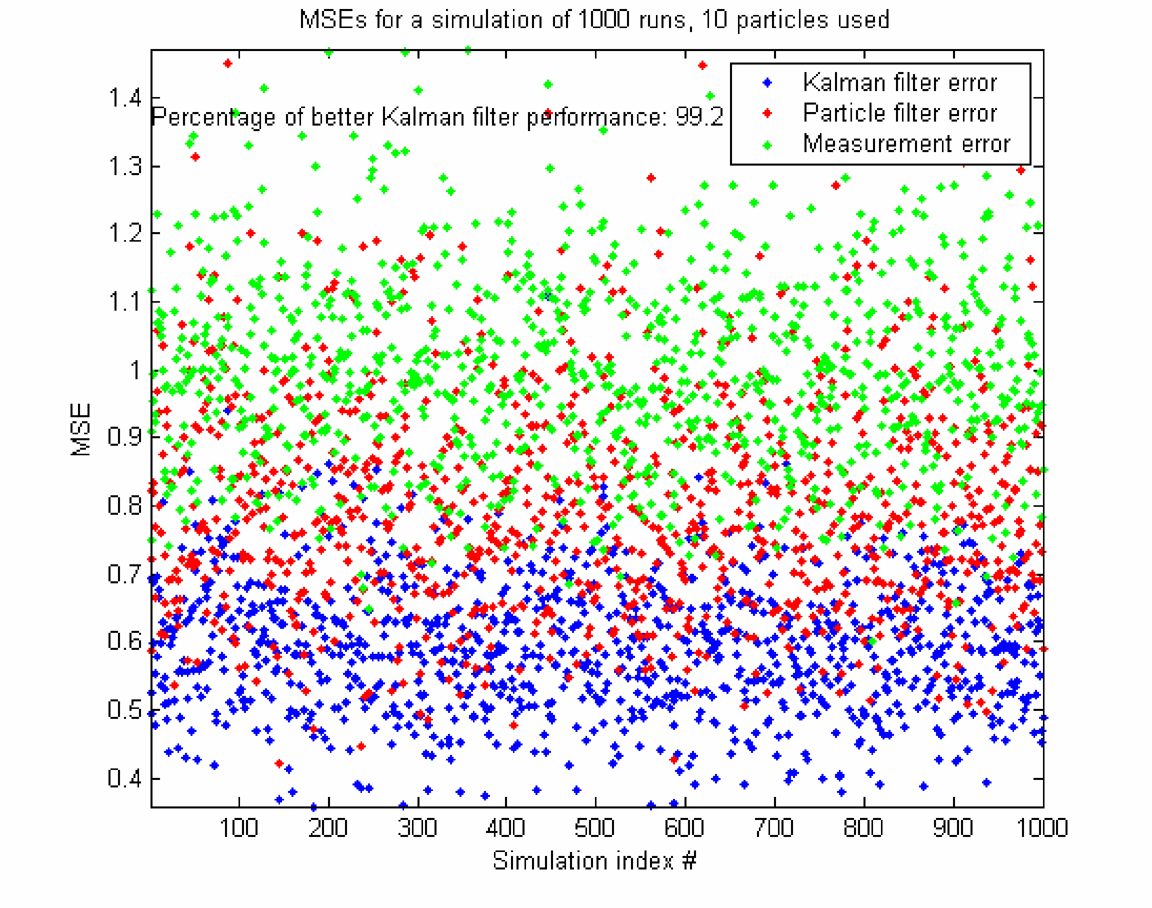
Although in this thesis the quaternion representation was used for all rotations, a better approach would be to represent small angular updates with the Gibbs vector and then project back to quaternion. The reason behind this logic is the fact that Gibbs vectors are additive and do not need normalization, while quaternions need to be normalized at every step, thus introducing possible biased estimates.

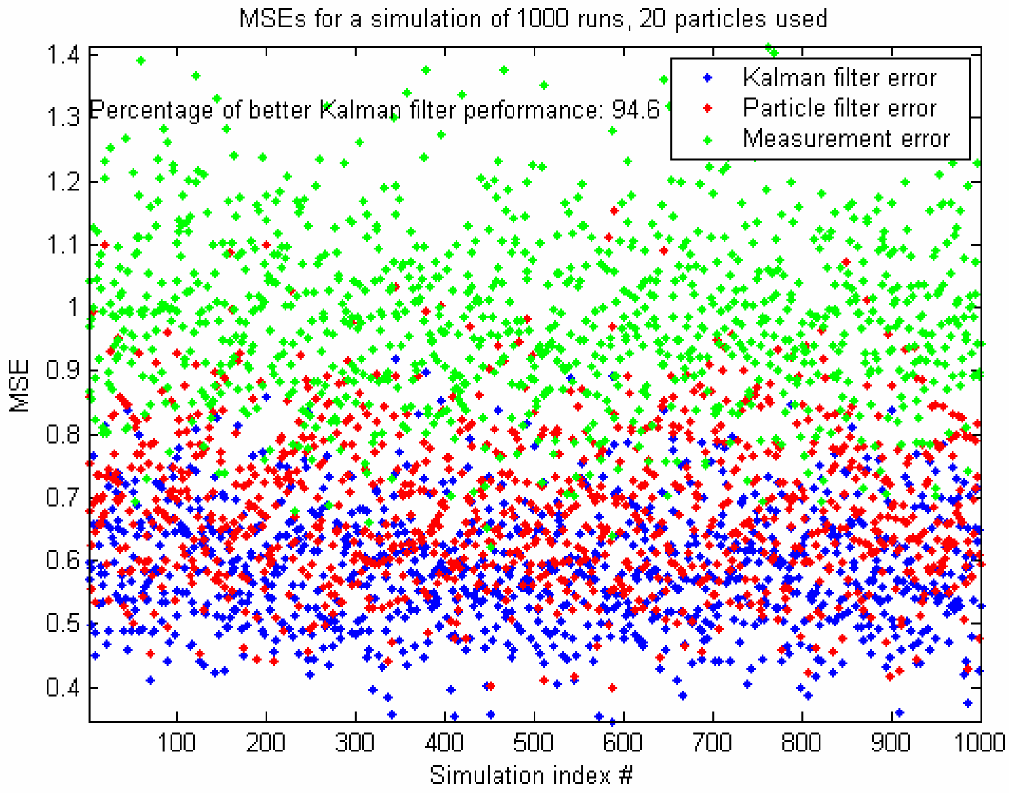
THIS PAGE INTENTIONALLY LEFT BLANK

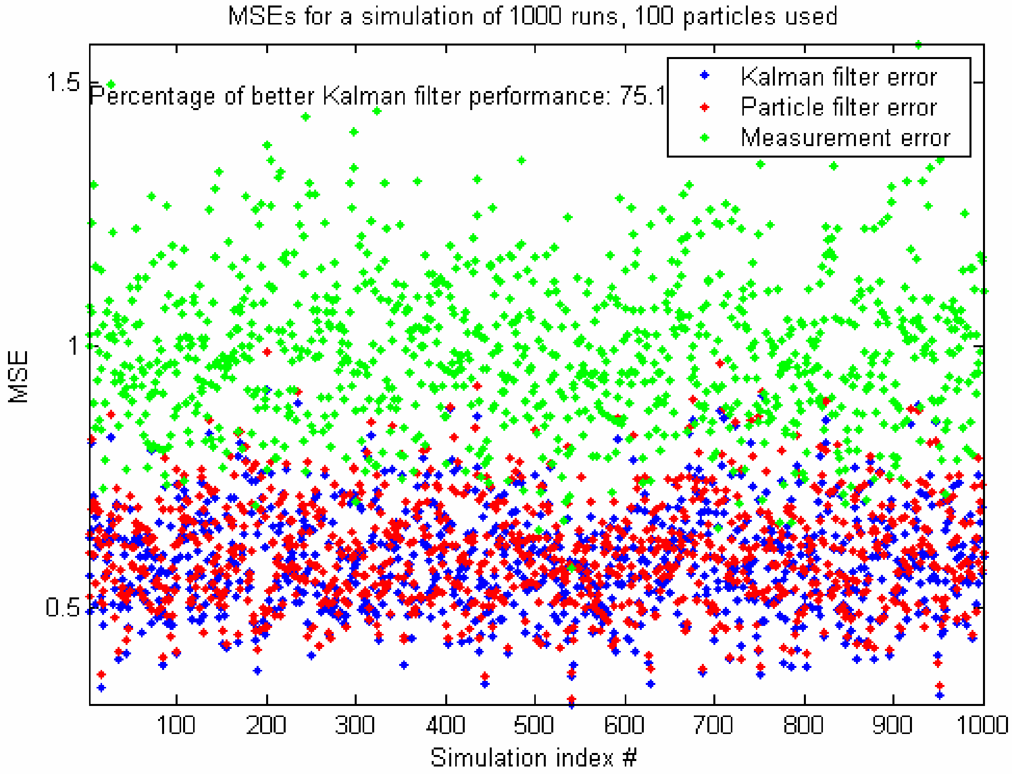
APPENDIX A: SCATTER PLOTS FOR THE MSE IN THE LINEAR MODEL

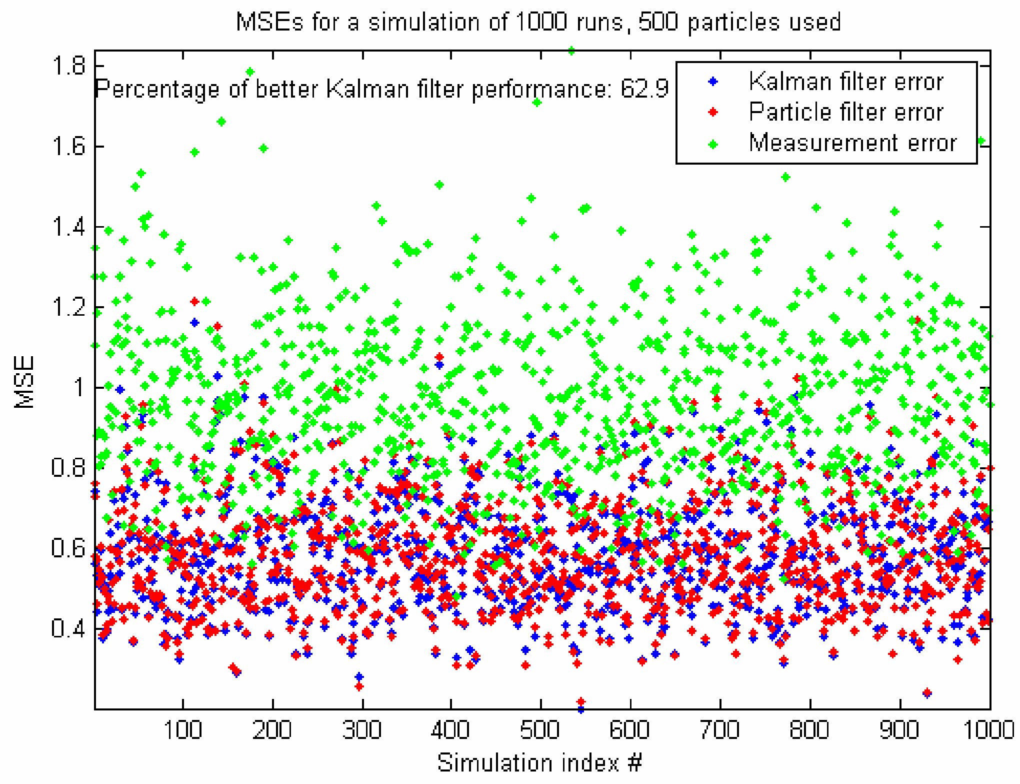
A. GAUSSIAN ADDITIVE NOISE CASE

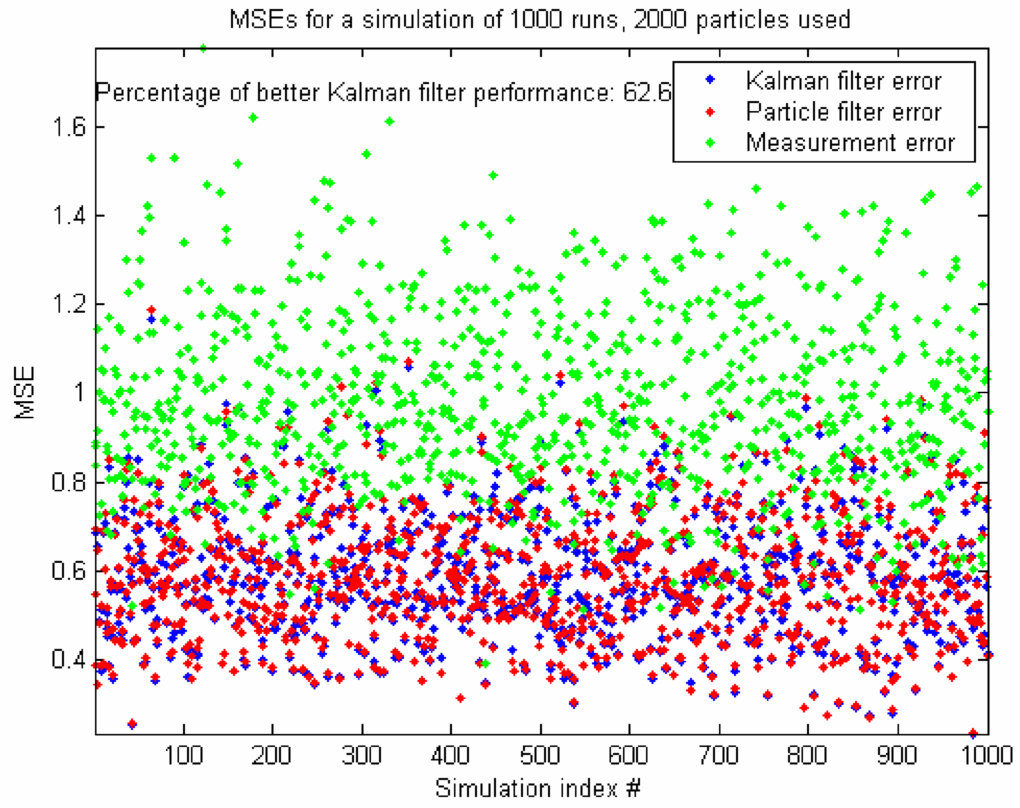
The following figures are associated with the results presented by the histograms in Figs. 5 through 10 of the thesis, for 10, 20, 100, 500, 2000 and 5000 particles. The exact value of the MSE is plotted as a single dot for every simulation and the comparison of the performance of the two filters can be visualized by the vertical distance of the related dots in the plot. It is clearly obvious that, when the number of particles is increased, the distance between the red (PF MSE) and the blue (KF MSE) dots is reduced significantly verifying the theoretical approach that the PF converges to the KF when infinite number of particles is used.

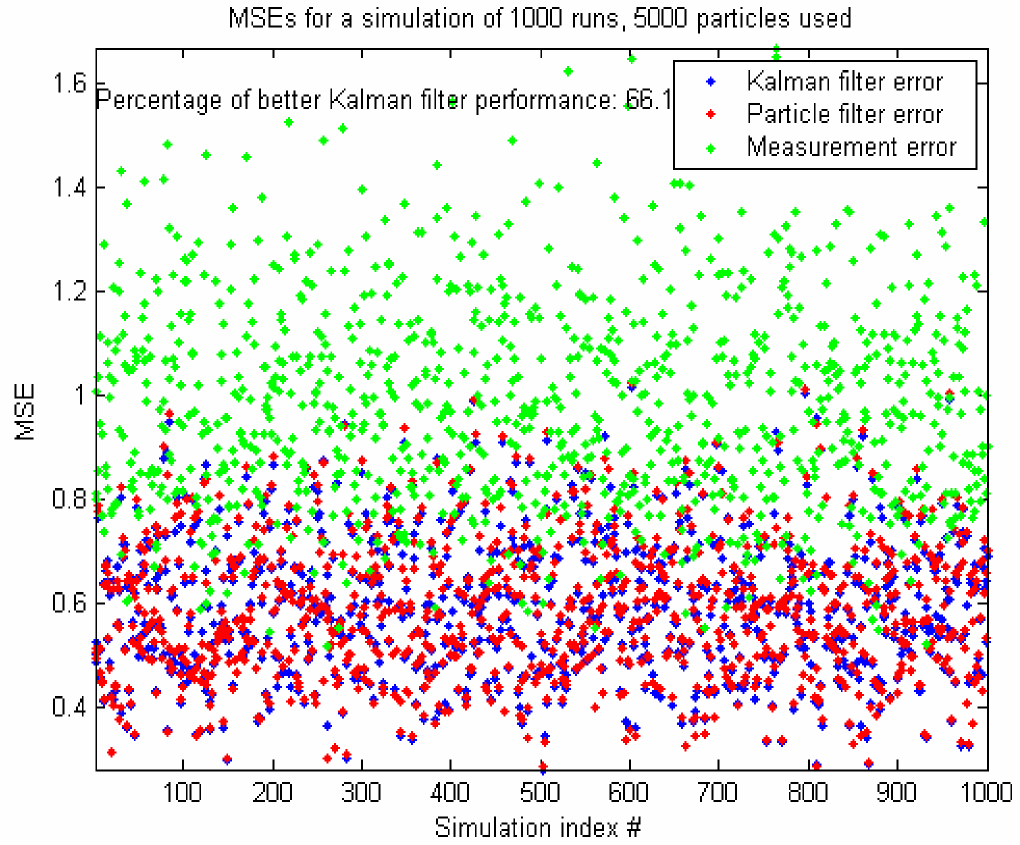



MSEs for a simulation of 1000 runs, 20 particles used
Percentage of better Kalman filter performance: 94.6
Kalman filter error
Particle filter error
Measurement error
MSE
1.4
1.3
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
100
200
300
400
500
600
700
800
900
1000
Simulation index #


MSEs for a simulation of 1000 runs, 100 particles used
Percentage of better Kalman filter performance: 75.1
Kalman filter error
Particle filter error
Measurement error
MSE
1.5
1
0.5
100
200
300
400
500
600
700
800
900
1000
Simulation index #

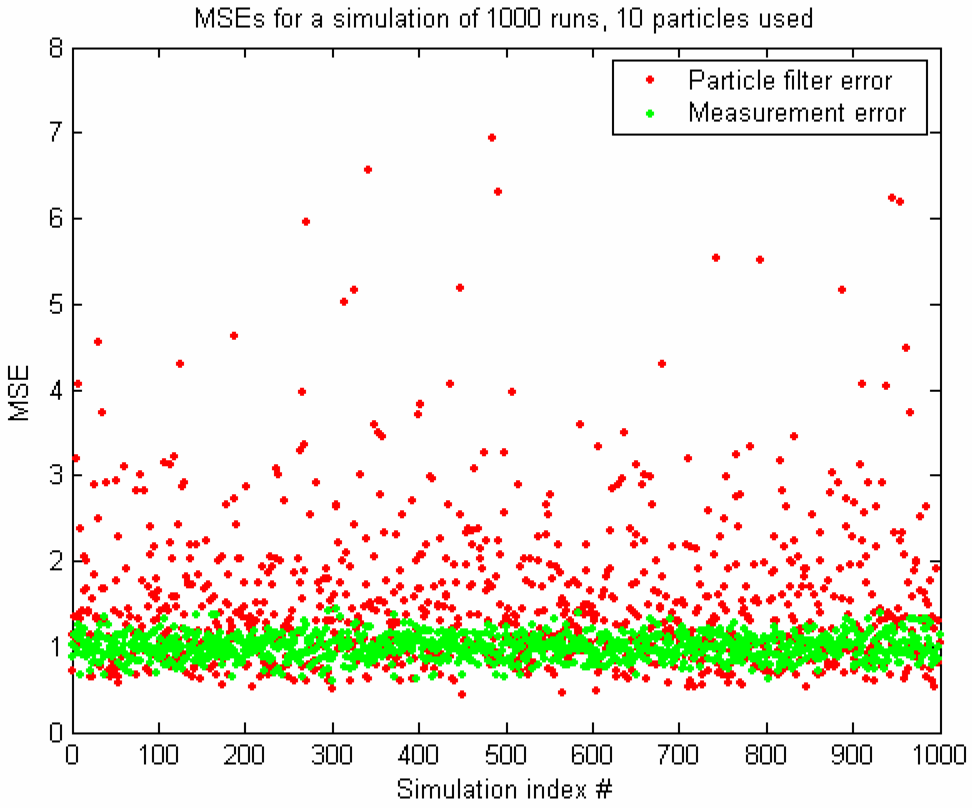

MSEs for a simulation of 1000 runs, 500 particles used
Percentage of better Kalman filter performance: 62.9
Kalman filter error
Particle filter error
Measurement error
MSE
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
100
200
300
400
500
600
700
800
900
1000
Simulation index #

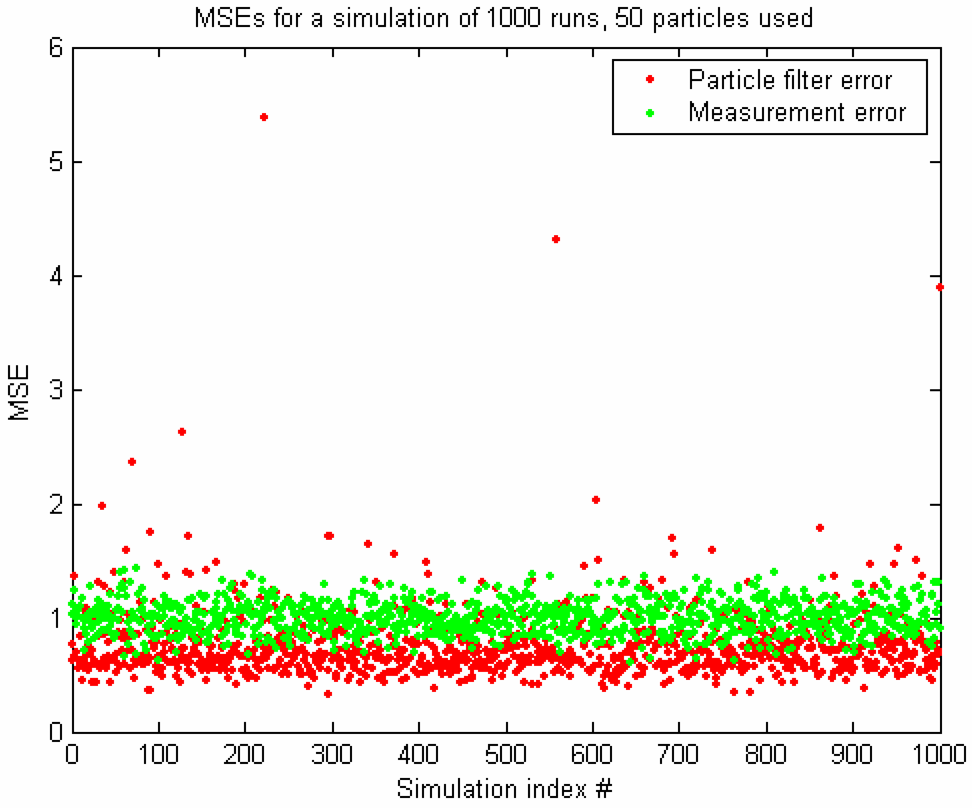

MSEs for a simulation of 1000 runs, 2000 particles used
Percentage of better Kalman filter performance: 62.6
Kalman filter error
Particle filter error
Measurement error
MSE
1.6
1.4
1.2
1
0.8
0.6
0.4
100
200
300
400
500
600
700
800
900
1000
Simulation index #

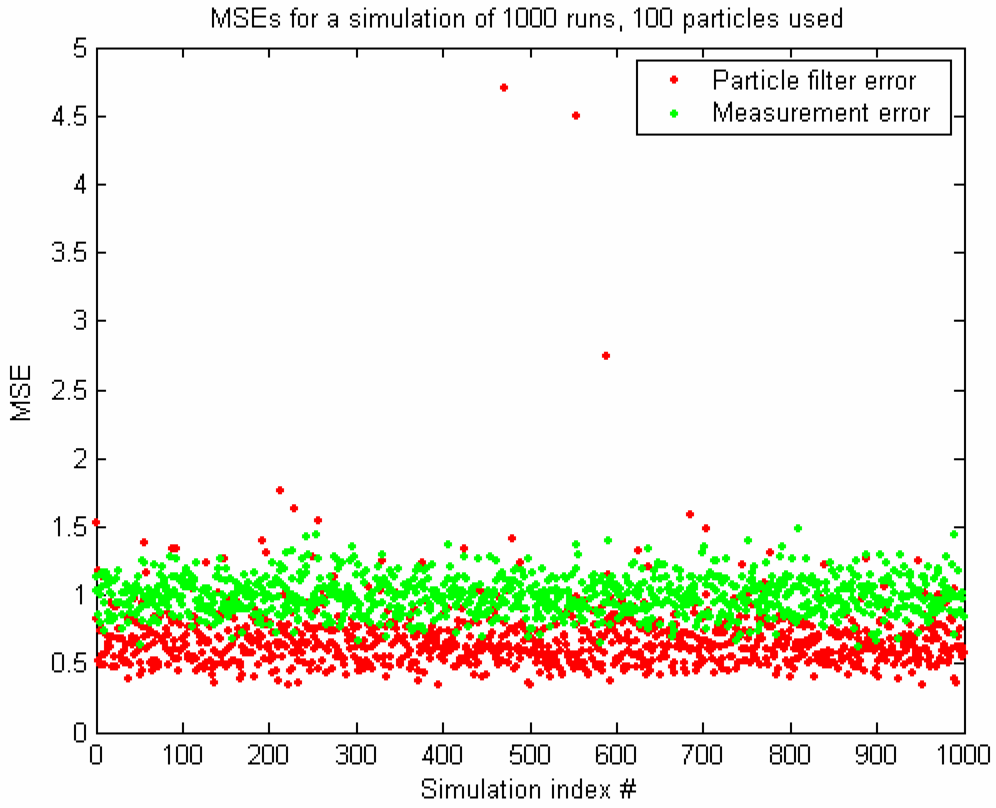


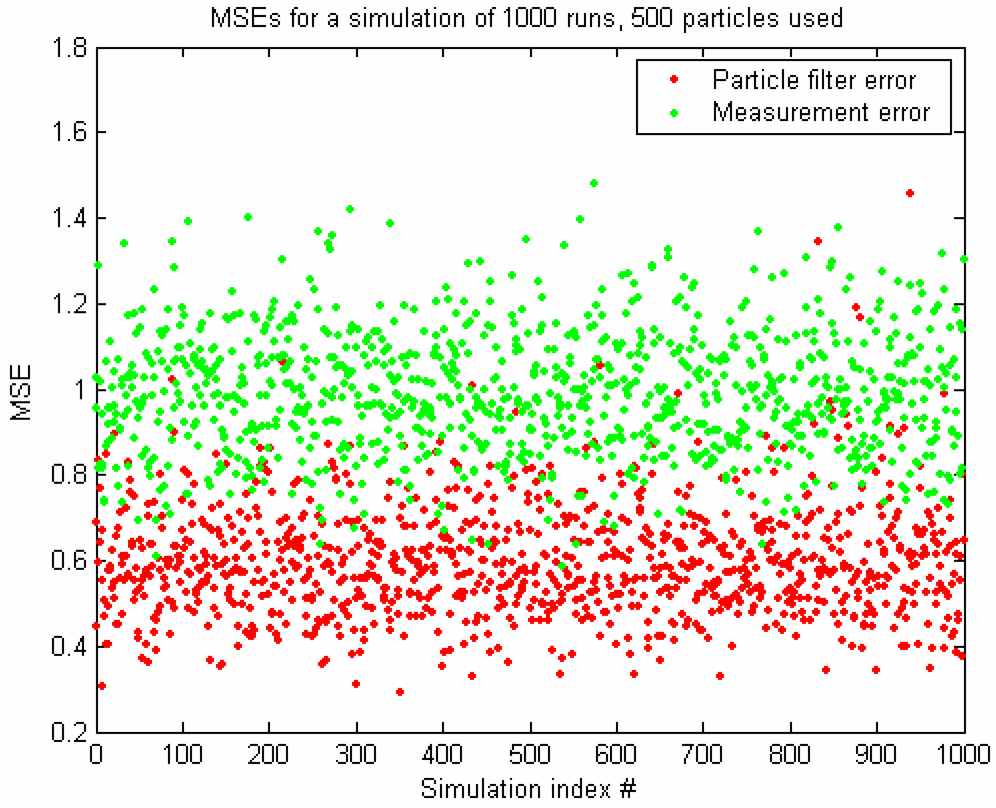
B. NON-GAUSSIAN ADDITIVE NOISE CASE

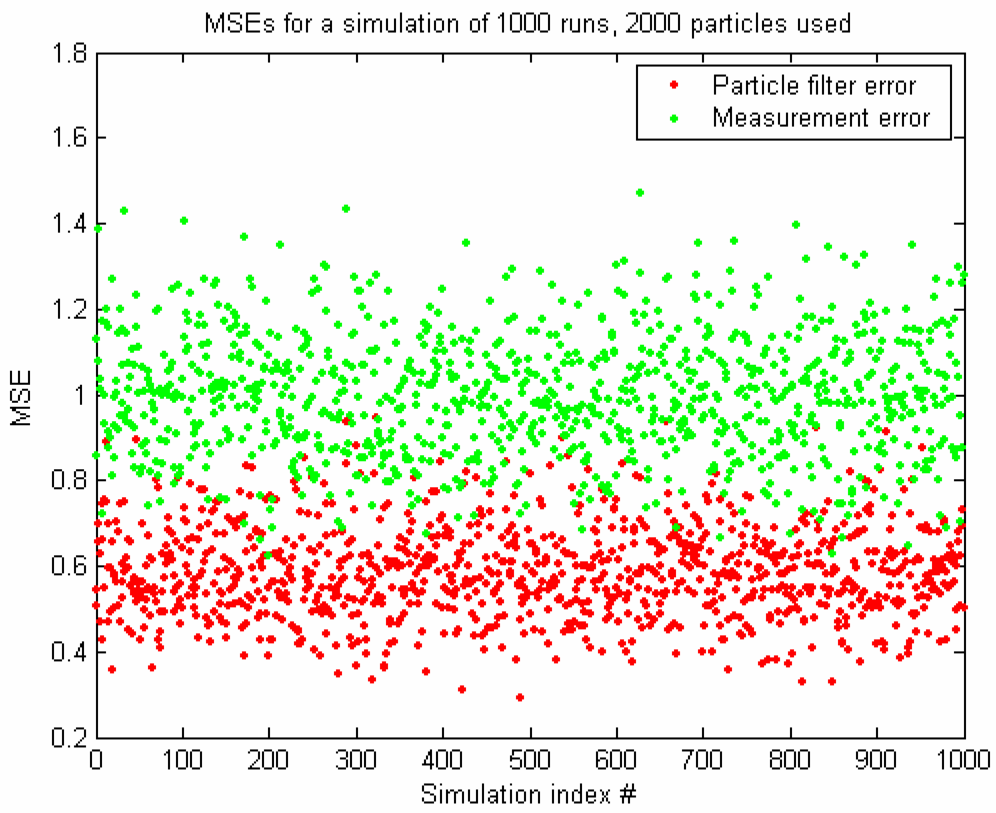
The following figures are associated directly to the results presented by the histograms in Figs. 11 through 15 of the thesis for 10, 50, 100, 500 and 2000 particles. The same mapping as the previous section of the appendix is used. With these plots it is quite obvious that in a non-Gaussian environment, an insufficient number of particles may result in a probable nonconvergence for the PF. This can be shown by the large values of MSE that are noticeable in a number of simulations. As the number of particles increases, the phenomenon is reduced and finally eliminated in the 2000-particle case.


MSEs for a simulation of 1000 runs, 10 particles used
MSE
Simulation index #
Particle filter error
Measurement error
0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700
800
900
1000


MSEs for a simulation of 1000 runs, 50 particles used
MSE
Simulation index #
Particle filter error
Measurement error
0
1
2
3
4
5
6
0
100
200
300
400
500
600
700
800
900
1000


MSEs for a simulation of 1000 runs, 100 particles used
● Particle filter error
● Measurement error
MSE
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
100
200
300
400
500
600
700
800
900
1000
Simulation index #


MSEs for a simulation of 1000 runs, 500 particles used
● Particle filter error
● Measurement error
MSE
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
0
100
200
300
400
500
600
700
800
900
1000
Simulation index #


MSEs for a simulation of 1000 runs, 2000 particles used
MSE
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
100
200
300
400
500
600
700
800
900
1000
Simulation index #
Particle filter error
Measurement error

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B: MATLAB® CODE USED FOR THE NONLINEAR MODEL APPLICATION

This following section includes the code that was used with the Matlab® software to perform the simulations for the nonlinear application. The code consists of the main module (called SIR_nonlinear.m), the functions called through it (SIR_fun.m, predict.m, weightvector.m and resample.m) and the Simulink® model (called quat2euler.mdl) which provides accurate conversion of the quaternions to Euler angles. Each one of the Matlab® files is accompanied by the appropriate comments for further reference. Some basic ideas for the functions SIR_fun.m, predict.m, weightvector.m and resample.m were found in Ref. [15].

A. MAIN MODULE

```
% LT IOANNIS KASSALIAS
% DATA GENERATION AND FILTERING ALGORITHM FOR THE
NONLINEAR PROBLEM
% Generate the true values of angular rates given the model
% omega=A*sin(omega_0*t+phi) for the 1 dimensional case
% Calculate the true values of quaternions and Euler angles
% Generate noise sequences for both angular rate and Euler angle
% measurement
% Calculate the related "noisy" values of the quaternion
% Perform SIR filtering and plot the results
clc
clear
format short g
tic

% Define parameters
A=10^0;                  % Max amplitude of the angular rate
omega_0=0.1;                 % Frequency used to generate angular rate
values
% phi=[0.1,0.2,0.3];         % Used as test for the initialization of the angu-
lar rate
phi=[0 0 0];             % Initial phase for the angular rate value
sigma=10^-1;                 % Standard deviation for the noise added to an-
gular rate
sigma2=10^1;                 % Standard deviation for the noise added to
Euler angles
q0=[0;0;0;1];             % Quaternion initial condition
Dt=0.01;                 % Sampling interval
t_final=20;              % Simulation duration
```

```
% Generate data for angular rates
t=Dt:Dt:t_final;                    % Time vector
for ii=1:3
   if ii==1
      omega_true(ii,:)=A*sin(omega_0*t+phi(ii));  % True value of omega
   else
      omega_true(ii,:)=zeros(1,t_final/Dt);
   end
end
plot(t,omega_true(1,:),t,omega_true(2,:),'r',t,omega_true(3,:),'g'),grid
title('True values of \omega')
xlabel('Time (sec)')
ylabel('\omega_t_r_u_e (rad/sec)')
legend('\omega_1','\omega_2','\omega_3',0)

% Generate OMEGA matrices of angular rates
for n=1:1:t_final/Dt
   omega_true_cross(:,:,n)=[0    -omega_true(3,n)    omega_true(2,n)    ;
omega_true(3,n) 0 -omega_true(1,n) ; -omega_true(2,n) omega_true(1,n)
0];
   OMEGA(:,:,n)=[-omega_true_cross(:,:,n)     omega_true(:,n)     ;     -
(omega_true(:,n)') 0];
end

% Compute quaternions
% Solve differential equation q_dot=0.5*OMEGA*q*Dt
% treating it as a difference equation in discrete time
% and considering a time interval Dt=0.01 sec
q_true(:,:,1)=q0;                  % Set initial condition
for n=2:1:t_final/Dt
   q_true(:,:,n)=Dt*0.5*OMEGA(:,:,n-1)*q_true(:,:,n-1)+q_true(:,:,n-1);
   q_true(:,:,n)=q_true(:,:,n)/norm(q_true(:,:,n));                 % Normalize
quaternions
end

% Plot the quaternions
q_1=reshape(q_true(1,1,:),1,t_final/Dt);
q_2=reshape(q_true(2,1,:),1,t_final/Dt);
q_3=reshape(q_true(3,1,:),1,t_final/Dt);
q_0=reshape(q_true(4,1,:),1,t_final/Dt);
figure
plot(t,q_1,t,q_2,'r',t,q_3,'g',t,q_0,'m'),grid
title('Plot of the quaternions vs. time')
xlabel('Time (sec)')
ylabel('Quaternion values (normalized)')
```

```
legend('q_1','q_2','q_3','q_0',0)

% Detect the zero crossings for quaternions
for n=2:1:t_final/Dt
   if sign(q_1(n+1))~=sign(q_1(n))
      tt1=n+1;
      break
   end
end
for n=tt1:1:t_final/Dt
   if sign(q_1(n+1))~=sign(q_1(n))
      tt2=n+1;
      break
   end
end

for n=2:1:t_final/Dt
   if sign(q_0(n+1))~=sign(q_0(n))
      tt3=n+1;
      break
   end
end
for n=tt3:1:t_final/Dt
   if sign(q_0(n+1))~=sign(q_0(n))
      tt4=n;
      break
   end
end

% Generate sign sequences in order to solve correctly the normalization
% constraint equation q1=+-sqrt(1-q0^2)
flag=ones(1,2000);
for i=tt3:tt4
   flag(i)=-flag(i);
end
flag2=ones(1,2000);
for i=tt1:tt2
   flag2(i)=-flag2(i);
end

% Conversion of quaternions to Euler angles
E_true=[];
for n=1:1:t_final/Dt
   q_true_0=q_true(1,1,n);                              q_true_1=q_true(2,1,n);
q_true_2=q_true(3,1,n); q_true_3=q_true(4,1,n);
   quaternion=[q_true_0 q_true_1 q_true_2 q_true_3];
   sim('Quat2Euler')
   E_true=[E_true,eul'];
end
figure
plot(t,E_true(1,:),t,E_true(2,:),'r',t,E_true(3,:),'g'),grid
title('True values of Euler angles versus time')
xlabel('Time (sec)')
ylabel('Euler angles (degrees)')
legend('Roll angle \phi','Pitch angle \theta','Yaw angle \psi',0)
E1=E_true(1,:);

% Generate measured value of angular rate sequences
omega_m=[];
for ii=1:3
   if ii==1
      n_w=sigma*randn(1,t_final/Dt);              % Generate noise sequence
      omega_m(ii,:)=omega_true(ii,:)+n_w;
   else
      omega_m(ii,:)=omega_true(ii,:);
   end
end

% Plot the true versus the measured values of angular rates
figure
subplot(311)
plot(t,omega_true(1,:),'LineWidth',1),hold on
plot(t,omega_m(1,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\omega_1 (rad/sec)')
legend('True value','Noisy measurement',0)
title('True versus measured values of angular rate')
subplot(312)
plot(t,omega_true(2,:),'LineWidth',1),hold on
plot(t,omega_m(2,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\omega_2 (rad/sec)')
legend('True value','Noisy measurement',0)
subplot(313)
plot(t,omega_true(3,:),'LineWidth',1),hold on
plot(t,omega_m(3,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\omega_3 (rad/sec)')
legend('True value','Noisy measurement',0)

% Add noise to the Euler angles to generate measurement sequences
for ii=1:3
   if ii==1
      n_E=sigma2*randn(1,t_final/Dt);     % Generate noise sequence
      E_m(ii,:)=E_true(ii,:)+n_E;
   else
      E_m(ii,:)=E_true(ii,:);
   end
end

% Plot the true versus the measured values of Euler angles
figure
subplot(311)
plot(t,E_true(1,:),'LineWidth',3),hold on
plot(t,E_m(1,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\phi (degrees)')
legend('True value','Noisy measurement',0)
title('True versus measured values of Euler angles')
subplot(312)
plot(t,E_true(2,:),'LineWidth',3),hold on
plot(t,E_m(2,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\theta angle (degrees))')
legend('True value','Noisy measurement',0)
subplot(313)
plot(t,E_true(3,:),'LineWidth',3),hold on
plot(t,E_m(3,:),'r:'),grid,hold off
xlabel('Time (sec)'),ylabel('\psi (degrees)')
legend('True value','Noisy measurement',0)

% Conversion back to quaternions
E_m_rad=E_m*pi/180;
c1=cos(E_m_rad(1,:)/2);
s1=sin(E_m_rad(1,:)/2);
c2=cos(E_m_rad(2,:)/2);
s2=sin(E_m_rad(2,:)/2);
c3=cos(E_m_rad(3,:)/2);
s3=sin(E_m_rad(3,:)/2);
c1c2=c1.*c2;
s1s2=s1.*s2;
q_0_m=(c1c2.*c3-s1s2.*s3).*flag;
q_1_m=(s1.*c2.*c3+c1.*s2.*s3).*flag;
q_2_m=c1c2.*s3+s1s2.*c3;
q_3_m=c1.*s2.*c3-s1.*c2.*s3;
figure
subplot(221)
plot(t,q_0,t,q_0_m,'r'),xlabel('Time (sec)')
title('q_0 true versus calculated from noisy Euler angles measurements
values')
subplot(222)
plot(t,q_1,t,q_1_m,'r'),xlabel('Time (sec)')
```

```
title('q_1 true versus calculated from noisy Euler angles measurements
values')
subplot(223)
plot(t,q_2,t,q_2_m,'r'),xlabel('Time (sec)')
title('q_2 true versus calculated from noisy Euler angles measurements
values')
subplot(224)
plot(t,q_3,t,q_3_m,'r'),xlabel('Time (sec)')
title('q_3 true versus calculated from noisy Euler angles measurements
values')
t1=t;

%%% FILTERING STAGE %%%

% Define parameters
N         = 2000;         % Number of iterations
t        = 1:1:N;        % Time step index
R         = sigma2^2;      % Measurement noise variance
Q         = sigma^2;       % Angular rate noise variance
numSamples = 2000;          % Number of particles used in each iteration

% Generate error sequences
v = E_true(1,:)-E_m(1,:);
w = q_1-q_1_m;
figure
subplot(211)
plot(t1,w),grid,xlabel('Time')
ylabel('Quaternion q_1 error')
subplot(212)
plot(t1,v),grid,xlabel('Time')
ylabel('Roll angle error in degrees')

% Determine state and measurement
x=q_1_m';                  % State sequence calculated by noisy angular rate
measurement
y=E_m(1,:)'*pi/180;          % Noisy roll angle measurement

% Perform SIR filtering
[samples,q] = SIR_fun(y,R,Q,numSamples,tt3,tt4);

% Compute MMSE estimate
q_1_est = mean(samples);

% Generate estimate of the scalar value quaternion
q_1_est=flag2.*abs(q_1_est);
q_0_est=flag.*abs(sqrt(1-q_1_est.^2));
```

```
% Plot filtered output for the quaternion
figure
subplot(211)
plot(t1,q_1,t1,q_1_est,'r'),xlabel('Time'),ylabel('Quaternion value q_1'),grid
legend('True value of q_1','Estimated value of q_1',0)
title('True versus estimated value of q_1')
subplot(212)
plot(t1,q_1,t1,q_1_m,'r'),xlabel('Time (sec)'),ylabel('Quaternion value
q_1'),grid
legend('True value of q_1','Value estimated by the noisy Euler angle
measurement')
title('q_1 true versus calculated from noisy Euler angles measurements
values')
figure
error_meas=abs(q_1-q_1_m);
error_filter=abs(q_1-q_1_est);
semilogy(t1,error_meas,t1,error_filter,'r')
xlabel('Time'),ylabel('Absolute error'),grid
legend('Error produced by noisy measurement','Error of the filtered out-
put',0)
title('Error representation in terms of quaternion value q_1')
err_rate_quat=mean(error_meas)/mean(error_filter)

% Conversion to Euler angles
E_est=[]; clear eul
for n=1:1:t_final/Dt
   quaternion=[q_1_est(n)  0   0   q_0_est(n)];
   sim('Quat2Euler')
   E_est=[E_est,eul'];
end
E2=E_est(1,:);

% Plot the true and the estimated values of the roll angle
figure
plot(t1,E1,t1,E2,'r'),grid
xlabel('Time'),ylabel('Roll angle in degrees')
legend('True value of roll angle','Estimated value of roll angle',0)
title('True versus the estimated from the filter output value of roll angle')

% Generate and plot the error curves
err_m_ang=abs(E1-E_m(1,:));
err_est_ang=abs(E1-E2);
semilogy(t1,err_m_ang,t1,err_est_ang,'r'),grid
xlabel('Time'),ylabel('Absolute error in  degrees')
legend('Initial error due to measurement','Error of the filtered output',0)
```

```
error_rate_angle=mean(err_m_ang)/mean(err_est_ang)
MSE_est=sum(err_est_ang.^2)/length(err_est_ang)
MSE_meas=sum(err_m_ang.^2)/length(err_m_ang)
toc
```

B. FUNCTION SIR_FUN.M

```
function [x,q] = SIR_fun(y,R,Q,numSamples,tt3,tt4)
% This function is used to perform SIR filtering for the specified
% nonlinear model. The given inputs are:
% y --> The available set of measurements
% R --> The measurement noise variance
% Q --> The process noise variance
% numSamples --> The number of particles
% tt3 --> The first zero crossing of quaternion q0
% tt4 --> The second zero crossing of quaternion q0
% The provided outputs are:
% x --> The estimated state trajectories
% q --> The normalized weighting vector

if nargin < 6, error('Not enough input arguments.'); end

[rows,cols] = size(y);              % rows = Number of iterations
x=zeros(numSamples,rows);            % Initialize posterior mean estimate
xu=zeros(numSamples,rows);           % Initialize predicted state
q=zeros(numSamples,rows);            % Initialize weighting vector

% Performing prediction and updating after resampling
for t=1:rows-1
  xu(:,t)  = predict(x(:,t),t,Q,tt3,tt4);
  q(:,t+1) = weightvector(xu(:,t),y(t+1,1),R,tt3,tt4);
  x(:,t+1) = resample(xu(:,t),q(:,t+1));
  x(:,t+1) = xu(:,t);
  [q(:,t+1),xu(:,t),x(:,t+1)];
end
```

C. FUNCTION PREDICT.M

```
function xu = predict(x,t,Q,tt3,tt4)
% This function provides the results of the prediction stage of the SIR
% based particle filter. The inputs are:
%  x --> The generated samples of the state
%  t --> The iteration index number
%  Q --> The variance of the process
% tt3 --> The first zero crossing of quaternion q0
% tt4 --> The second zero crossing of quaternion q0

if nargin < 5, error('Not enough input arguments.'); end

w  = sqrt(Q)*randn(size(x));
```

```
if (t>=tt3)&&(t<=tt4)
   xu = -0.5.*(sin(0.1*t*0.01)+w).*sqrt(1-x.^2)*0.01+ x;
else
   xu = 0.5.*(sin(0.1*t*0.01)+w).*sqrt(1-x.^2)*0.01+ x;
end
```

D. FUNCTION WEIGHTVECTOR.M

```
function q = weightvector(xu,y,R,tt3,tt4)
% This function computes the normalized weighting vector q
% The given inputs are:
%  xu --> The set of state prediction samples
%   y --> The available set of measurements
%   R --> The measurement noise covariance
% tt3 --> The first zero crossing of quaternion q0
% tt4 --> The second zero crossing of quaternion q0
% NOTE: The equation for variable m is the measurement equation
% for the one-axis case
if nargin < 5, error('Not enough input arguments.'); end

flag=ones(1,2000);
for i=tt3:tt4
   flag(i)=-flag(i);
end
[rows,cols] = size(xu);
q = zeros(size(xu));
m = atan2(2*xu.*sqrt(1-xu.^2) , 1-2*xu.^2);
q = exp(-.5*R^(-1)*(y.*ones(size(xu))-m).^(2));
q = q/sum(q);
```

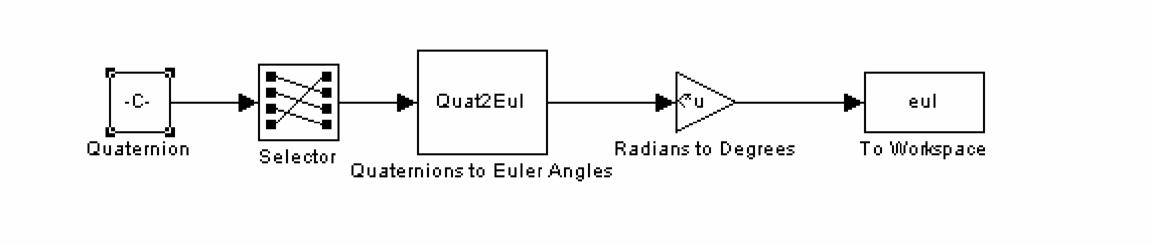
E. FUNCTION RESAMPLE.M

```
function x = resample(xu,q)
% This function performs the resampling stage providing the new set of
% the particles at the end of each iteration. The inputs are:
% xu --> The set of state prediction samples
%  q --> The normalized weighting vector

if nargin < 2, error('Not enough input arguments.'); end

i = 1;
j = 1;
[N,M]=size(xu);
u = rand(N+1,1);
x = ones(size(xu));
S = cumsum(-log(u));
W = cumsum(q);
while j <= N
  if (W(j,1)*S(N,1)) > S(i,1)
    x(i,1) = xu(j,1);
    i = i+1;
  else
    j = j+1;
  end
end
```

F. SIMULINK® MODEL QUAT2EULER.MDL



LIST OF REFERENCES

1. Robert G. Brown and Patrick Y.C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, John Wiley and Sons, Inc., New York, 1997.
2. B. Ristic, M.S. Arulampalam and N. Gordon, *Beyond the Kalman Filter Particle Filters for Tracking Applications*, Artech House, Boston, 2004.
3. M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," *IEEE Transactions on Signal Processing,* Vol. 50, No. 2, pp. 174-188, February 2002.
4. S. Haykin, K. Huber and Z. Chen, "Bayesian Sequential State Estimation for MIMO Wireless Communications," *Proceedings of the IEEE,* Vol. 92, No. 3, pp. 439-454, March 2004.
5. M.R. Morelande, S. Challa and N. Gordon, "Application of Particle Filters to Single Target Tracking Problems," *Signal and Data Processing of Small Targets 2003*, San Diego, CA, Aug. 5-7, 2003.
6. K. Dahia, C. Musso and D.T. Pham, "Application of the Kalman Filter to the Updated Inertial Navigation System," *ONERA: Tire a Part,* No. 179, pp. 1-4, 2004.
7. Y. Cheng and J.L. Crassidis, "Particle Filtering for Sequential Spacecraft Attitude Estimation," *AIAA Guidance, Navigation, and Control Conference and Exhibit,* Providence, Rhode Island, Aug. 16-19, 2004.
8. J.L. Crassidis and J.L. Junkins, *Optimal Estimation of Dynamic Systems,* CRC Press, Boca Raton, FL, 2004.
9. J.L. Crassidis and F.L. Markley, "Unscented Filtering for Spacecraft Attitude Estimation," *Journal of Guidance, Control and Dynamics*, Vol. 26, No. 4, pp. 536-542, July-August 2003.
10. J.L. Crassidis and F.L. Markley, "Sliding Mode Using Modified Rodrigues Parameters," *Journal of Guidance, Control and Dynamics*, Vol. 19, No. 6, pp. 1381-1383, November-December 1996.
11. F.L. Markley, "Attitude Error Representations for Kalman Filtering," *Journal of Guidance, Control and Dynamics,* Vol. 26, No. 2, pp. 311-317, March-April 2003.

12. J.S. Wertz, *Spacecraft Attitude Determination and Control,* D.Reidel Publishing Co., Dordrecht, The Netherlands, 1984.

13. J. Kim, R. Cristi and B. Agrawal, “Attitude Determination for NPS Three-Axis Spacecraft Simulator,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Providence, Rhode Island, Aug. 16-19, 2004.

14. M. Romano and B.N. Agrawal, “Acquisition, Tracking and Pointing of Bifocal Relay Mirror Spacecraft,” *Acta Astronautica*, Vol. 53, No. 4, pp. 509-519, February 2003.

15. Nando de Freitas Software, *Sequential Sampling-Importance Resampling (SIR)*. http://www.cs.ubc.ca/~nando/software.html/demo1.tar (last accessed April 2005).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Department Chairman, Code EC/Po
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

4. Department Chairman, Code IS/Bo
   Department of Information Sciences
   Naval Postgraduate School
   Monterey, California

5. Professor Roberto Cristi, Code EC/Cr
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California

6. Professor David C. Jenn, Code EC/Jn
   Department of Electrical and Computing Engineering
   Naval Postgraduate School
   Monterey, California

7. Embassy of Greece, Naval Attaché
   Washington, DC

8. Ioannis Kassalias
   11142, Athens
   GREECE